RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

University of Virginia

Charlottesville, Virginia 22901

Final Report

REAL TIME FLIGHT SIMULATION METHODOLOGY

NASA Grant No. NSG-1151-1

Submitted to:

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Submitted by:

E. A. Parrish
Associate Professor

G. Cook
Professor

E. S. McVey
Professor

Report No. UVA/528085/EE77/104

April 1977

Final Report

REAL TIME FLIGHT SIMULATION METHODOLOGY

NASA Grant No. NSG-1151-1

Submitted to:

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Submitted by:

E. A. Parrish
Associate Professor

G. Cook
Professor

E. S. McVey
Professor

Department of Electrical Engineering
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/528085/EE77/104
April 1977

Copy No. 1

FOREWORD

This report summarizes the current status and results achieved during the past year on research on Real-Time Flight Simulation Methodology. This work is a continuation of research performed during the first year on substitutional methods for digitization, input signal-dependent integrator approximations, and digital autopilot design.

The recent Semi-Annual Report [1] was extensive and covered, in detail, the first six months of effort. Consequently, that report is heavily referenced herein; and only the information necessary to provide proper perspective is repeated.

I. AN INTERACTIVE SIMULATOR DESIGN PACKAGE FOR THE DESIGN OF REAL-TIME SIMULATORS

1.0 Introduction

This section describes the status of research on an interactive software support system which will aid the design of optimum simulation models. The generic type of system under study is shown in Fig. 1.0. When programming is completed, it is envisioned that the Simulator Design Package (SDP) can be used to evaluate a number of different standard integrator models (for example, Tustin, Optimum Discrete Approximation) on the basis of selectable error criteria or design an entirely new model suitable for a particular problem. In the latter case the model would be designed on an interactive basis, using selectable algorithms to find an optimal form.

In previous work we examined a number of different substitution methods to determine which was most suitable under various error criteria. Based on these results, a number of substitution formulas have been chosen for inclusion in SDP. Consequently, most of the work during the first six months concentrated on the evaluation of optimization algorithms and discrete representations. The second six months have been used to develop a substantial amount of the software framework of SDP. This includes subroutines for iterative designs of simulation models as well as a rudimentary graphics package.

This section contains three sub-sections which report on different facets of the development of SDP. In the first sub-section we are concerned with assuming a discrete representation for a given continuous transfer function and then iterating to solve for optimal values of the parameters concerned. Preliminary results of time domain optimization are also discussed. In the second sub-section a similar effort is reported in which a form is assumed for an

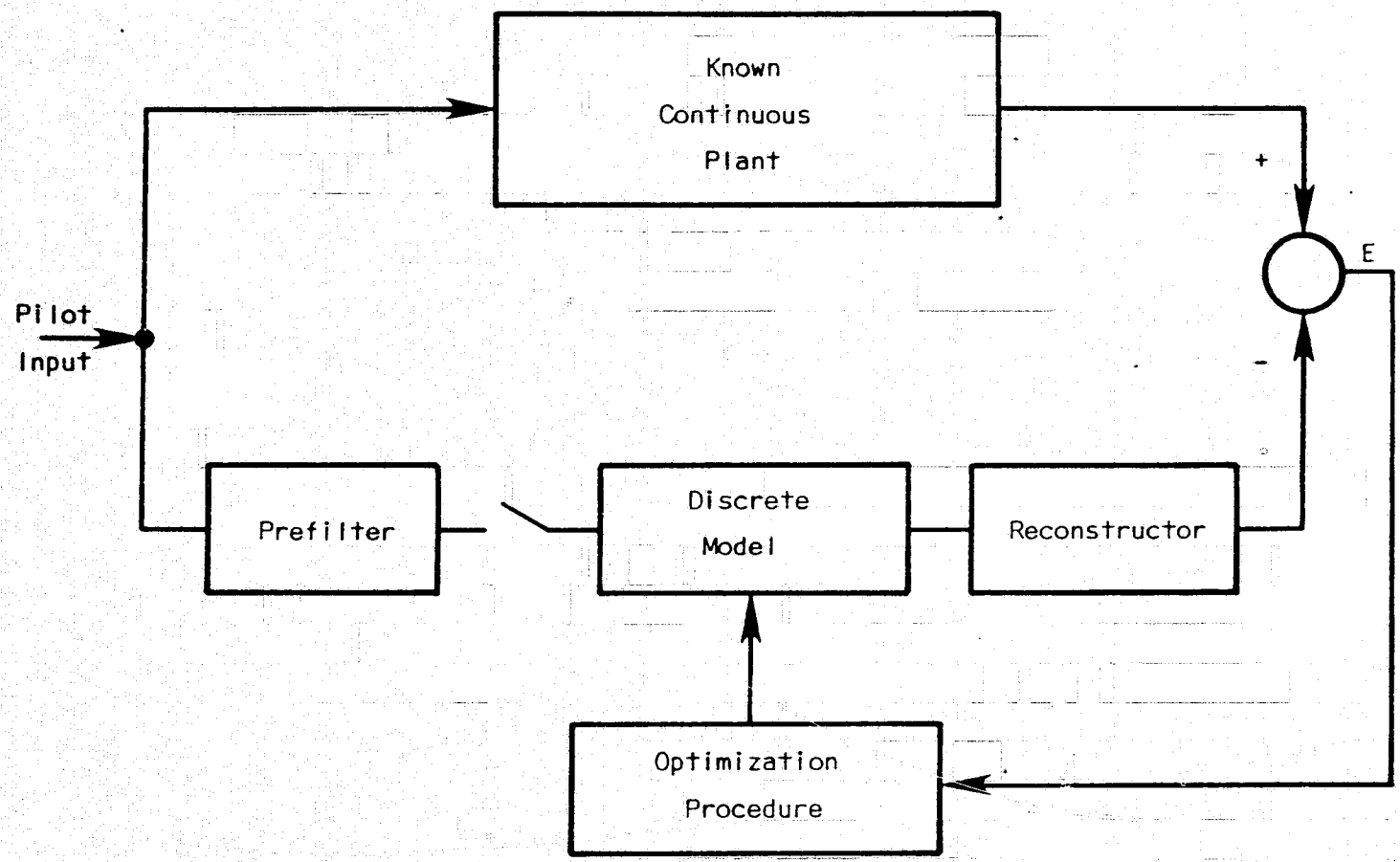


Figure 1.0 System Configuration for Model Development

integration operator, and then a random search method is used to determine the optimum parameters. In the third sub-section examples of graphical results obtained with SDP are presented.

1.1.1 Frequency Domain Optimization

As indicated in previous reports, attempts have been made to obtain a digital simulation model in a way that minimizes the frequency-domain error. We have discussed the general form of the digital system, the error criteria, and the gradient technique used. Implicit formuli for gradients were derived so that programs can be written to evaluate gradients necessary for the numerical technique being used.

Referring to the 1976 Semi-Annual Report [1], we have the following results:

Error:

$$E = \sum_{m=1}^{M} \left| H(j\omega_m) - \bar{H}(j\omega_m) \right|^2 \qquad (1\text{-}1)$$

Form of digital system:

$$H(z) = A \frac{\prod_{k=1}^{K} (1 + a_{2k-1} z^{-1} + a_{2k} z^{-2})(1 + a^1_{NZ} z^{-1})}{\prod_{k=1}^{NCP/2} (1 + 2b_{2K} \cos 2b_{2k-1} z^{-1} + b^2_{2k} z^{-2}) \prod_{k=NCP+1}^{NP} (1 + b_k z^{-1})} \qquad (1\text{-}2)$$

where NZ ≡ number of zeros

NP ≡ number of poles

NCP ≡ number of complex poles

k ≡ largest integer NZ/2

a^1_{NZ} = 0 if NZ = 2k(even number of zeros)

$$a_{NZ}^{1} \neq 0 \text{ if } NZ - 2k + 1 \text{(odd number of zeros)}$$

We also discussed the constraints necessary to limit the time domain error and to ensure stability.

Three FORTRAN programs have been written to accomplish the optimization in which the Fletcher-Powell method [2] was used. Since the algorithm is not critically important and the Fletcher-Powell method has proved to be successful, no other numerical methods were investigated. The process of obtaining the optimal digital form will be implemented off-line; so the speed-versus-accuracy criterion is no longer important in choosing the numerical algorithm. The Fletcher-Powell method converges rapidly in this particular application.

Application of the Frequency Optimization (FO) Method

Only recently were we able to use our programs to perform some preliminary comparisons. This is due partly to the complexity involved in programming. The programs were written in such a fashion that they can be incorporated into the SDP in the future. Only first-order systems have been investigated, to date. Higher-order systems will be considered in future reports. The first-order system under study is:

$$H(s) = \frac{1}{s + 1} \tag{1-3}$$

The selected sampling interval is $T = \Pi/10$ secs. This is rather large, but it will provide a good comparison with the Tustin method for large sampling intervals. Most existing simulation methods work fairly well with small sampling periods but suffer severely at larger sampling periods.

The frequency response is given by

$$H(j\omega) = \frac{1 - j\omega}{1 + \omega^2} \tag{1-4}$$

The actual response of this system will be compared with the responses of the simulation system for a step, ramp, and sinusoidal inputs.

Several different optimized systems were obtained and their performances compared to the above system. A distinct characteristic of this optimization procedure is that the user can arbitrarily specify the order of the simulation system, i.e., one can obtain a first-, second-, or third-order digital model for the simulated first-order system. As results will show, it is usually better to use the digital model of the same, or one order higher, than the continuous system. Furthermore, if the continuous system's poles are all real, then the user should choose his digital model to have all real poles in the z-plane. This is intuitively obvious, since all real poles in the s-plane map into real poles in the z-plane.

As discussed in the semi-annual report, in order to limit the time domain error we must place constraints on the pole locations of the digital system in the following fashion:

$$\text{z-plane poles} = \exp[T(\text{s-plane poles})] \tag{1-5}$$

or in a less restricted form:

$$|z| = |e^{sT}| = e^{T(\text{Real } s)} \tag{1-6}$$

This form determines the radius of a circle (in the z-plane). All poles (both real and complex) must be constrained to lie within this circle. We also allow this radius to approach unity in order to show that the constraint does, indeed, reduce time-domain error.

The following digital systems are models of the first-order continuous system which we simulate:

(A) Poles of digital systems are restricted to lie within the circle of radius $e^{-T} = .730403$

1st-Order Model:

$$H(z) = K \frac{1 + .986629z^{-1}}{1 - .713722z^{-1}} \tag{1-7}$$

where K = 1.144102

2nd-Order Model:

$$H(z) = K \frac{1 + 2.42315z^{-1} + .518598z^{-2}}{(1 - .729328z^{-1})(1 + .730403z^{-1})} \tag{1-8}$$

where K = .118823

(B) Poles of digital systems are restricted to lie within the circle of radius .9

1st-Order Model:

$$H(z) = K \frac{1 + .986629z^{-1}}{1 - .713722z^{-1}} \tag{1-9}$$

where K = .144102

2nd-Order Model:

$$H(z) = K \frac{1 + 2.78393z^{-1} + .743926z^{-2}}{(1 - .728308z^{-1})(1 + .90000z^{-1})} \tag{1-10}$$

where K = .114034

(C) Poles of digital systems are restricted to lie within the circle of radius .999

1st-Order Model:

$$H(z) = K \frac{1 + .986629z^{-1}}{1 - .713722z^{-1}} \tag{1-11}$$

where K = .144102

2nd-Order Model:

$$H(z) = K \frac{(1 + 3.01207z^{-1} + .885306z^{-2})}{(1 - .727803z^{-1})(1 + .999z^{-1})} \tag{1-12}$$

where K = .111105

All K are chosen so that the steady-state gain is 1, which is the steady-state gain of the continuous system.

Notice that all first-order models are the same, regardless of the constraint. This is possible, because we are simulating a first-order system. For second-order models the constraint plays a more important role. The optimization algorithm tends to place poles as far as possible within the constraint. This results in poles at -.9 (for $|\text{Poles}| \leq .9$ constraint) and at -.999 (for $|\text{Poles}| \leq .999$ constraint). These poles dominate the response of the system and, as a result, give more error.

The following figures show some of the advantages and disadvantages of the FO method. The method is far better than Tustin, especially at high sampling periods. As stated before, the digital systems are designed for a sampling period of $T = \Pi/10$ secs. This is rather large for the Tustin method, whose

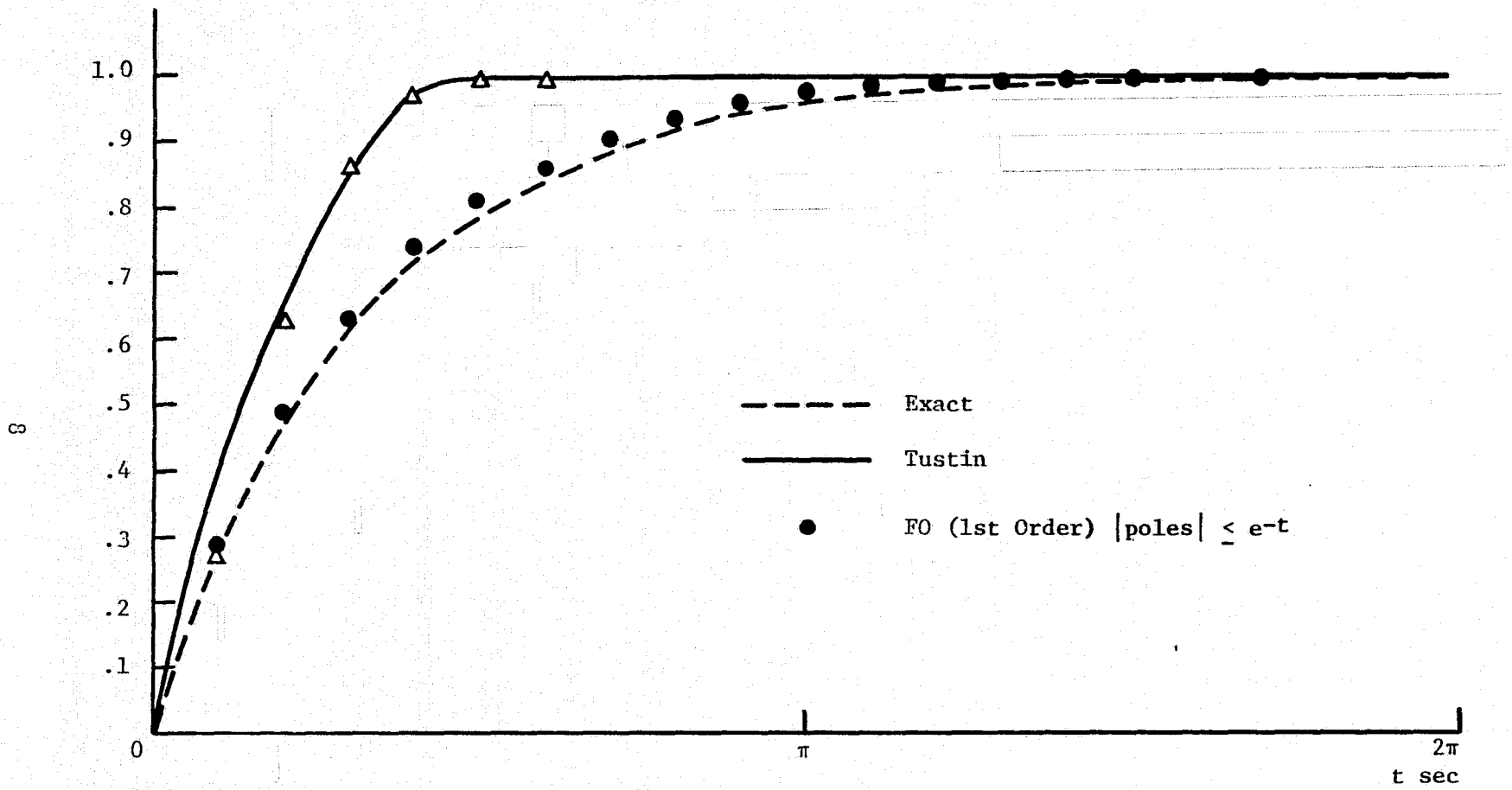


Figure 1-1 Comparison of Exact, Tustin, and 1st Order FO Responses

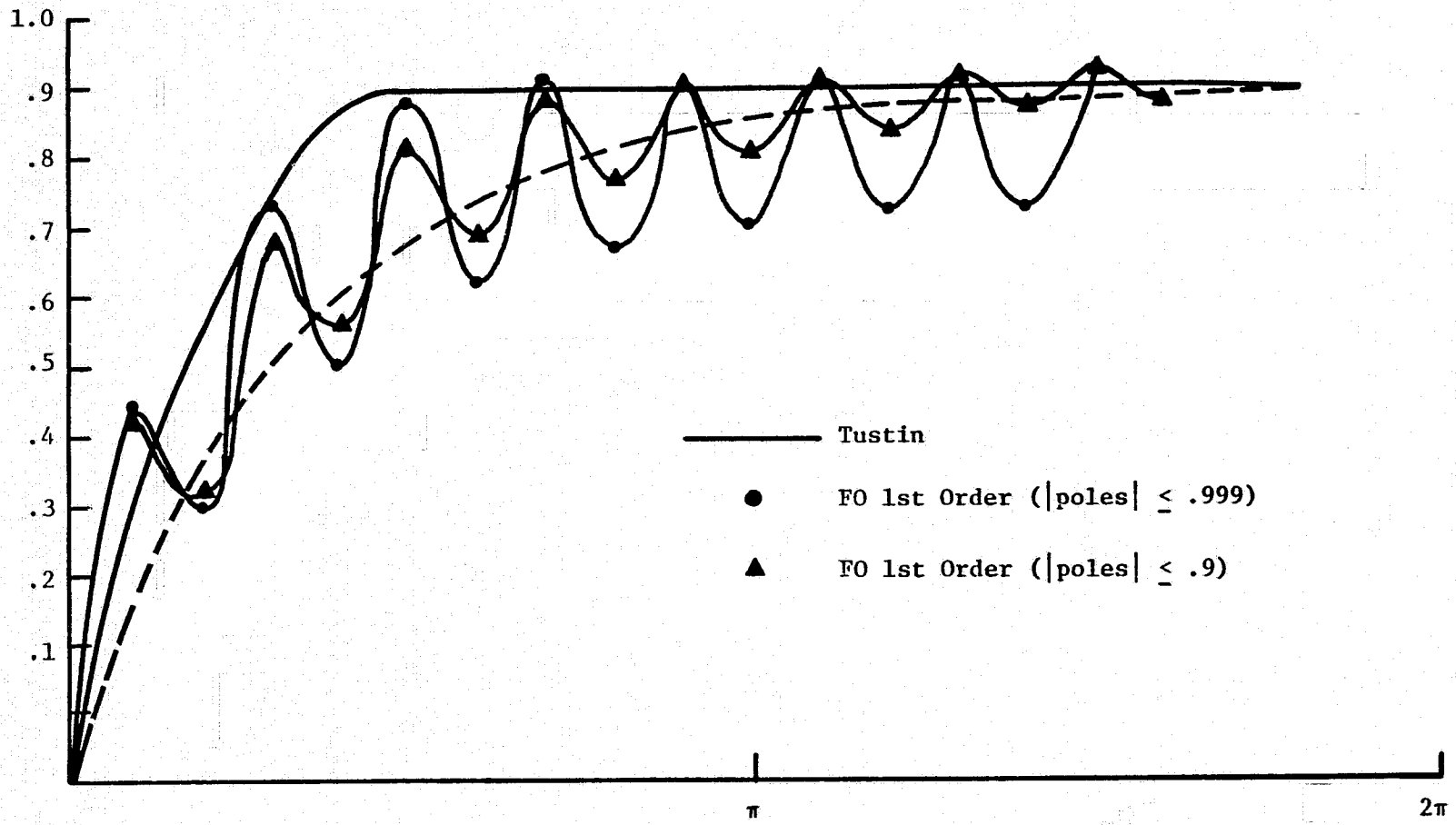


Figure 1-2 Comparison of Exact, Tustin, and 1st Order FO Responses (Constraint Relaxed)

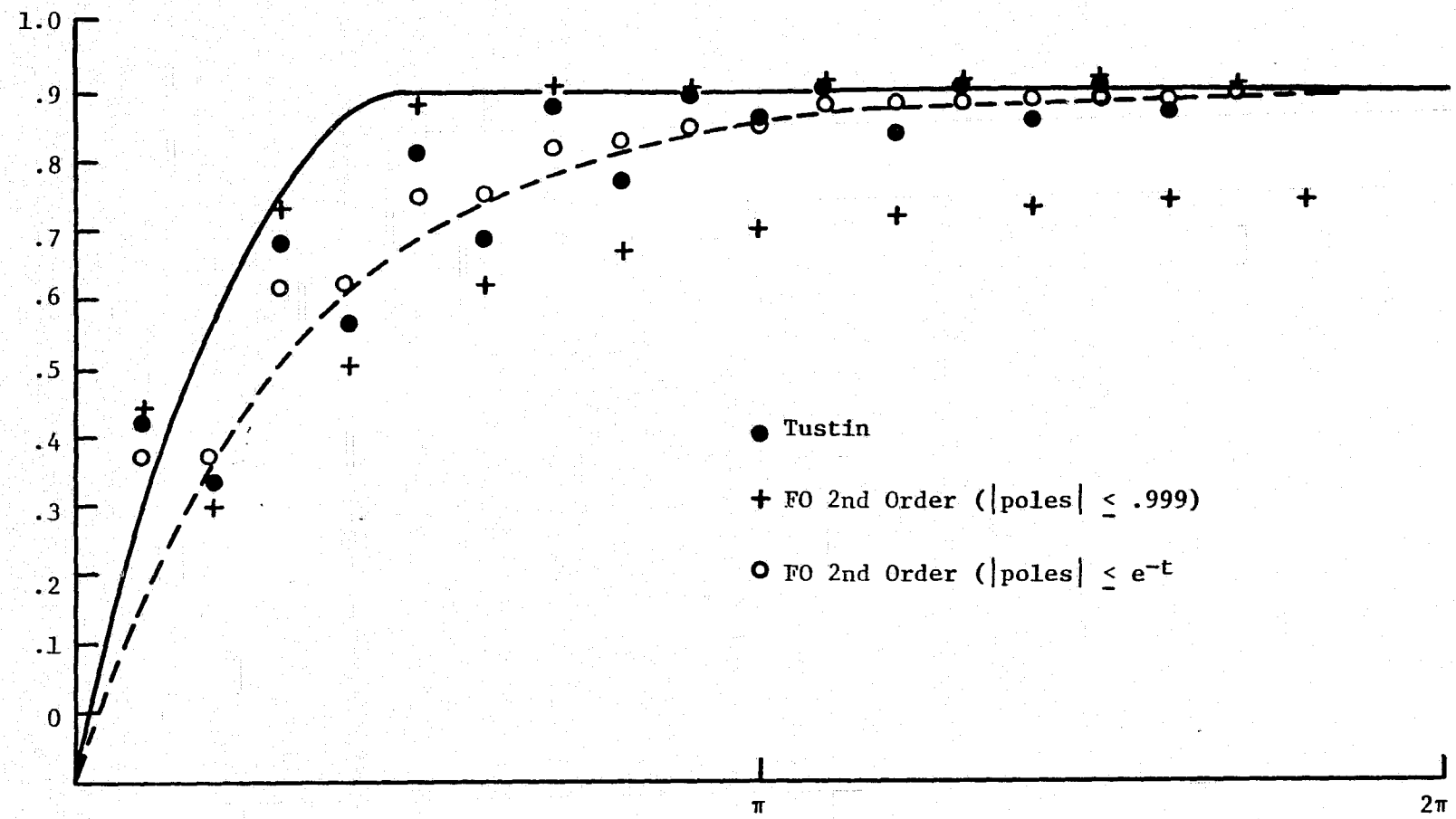


Figure 1-3 2nd Order FO Responses

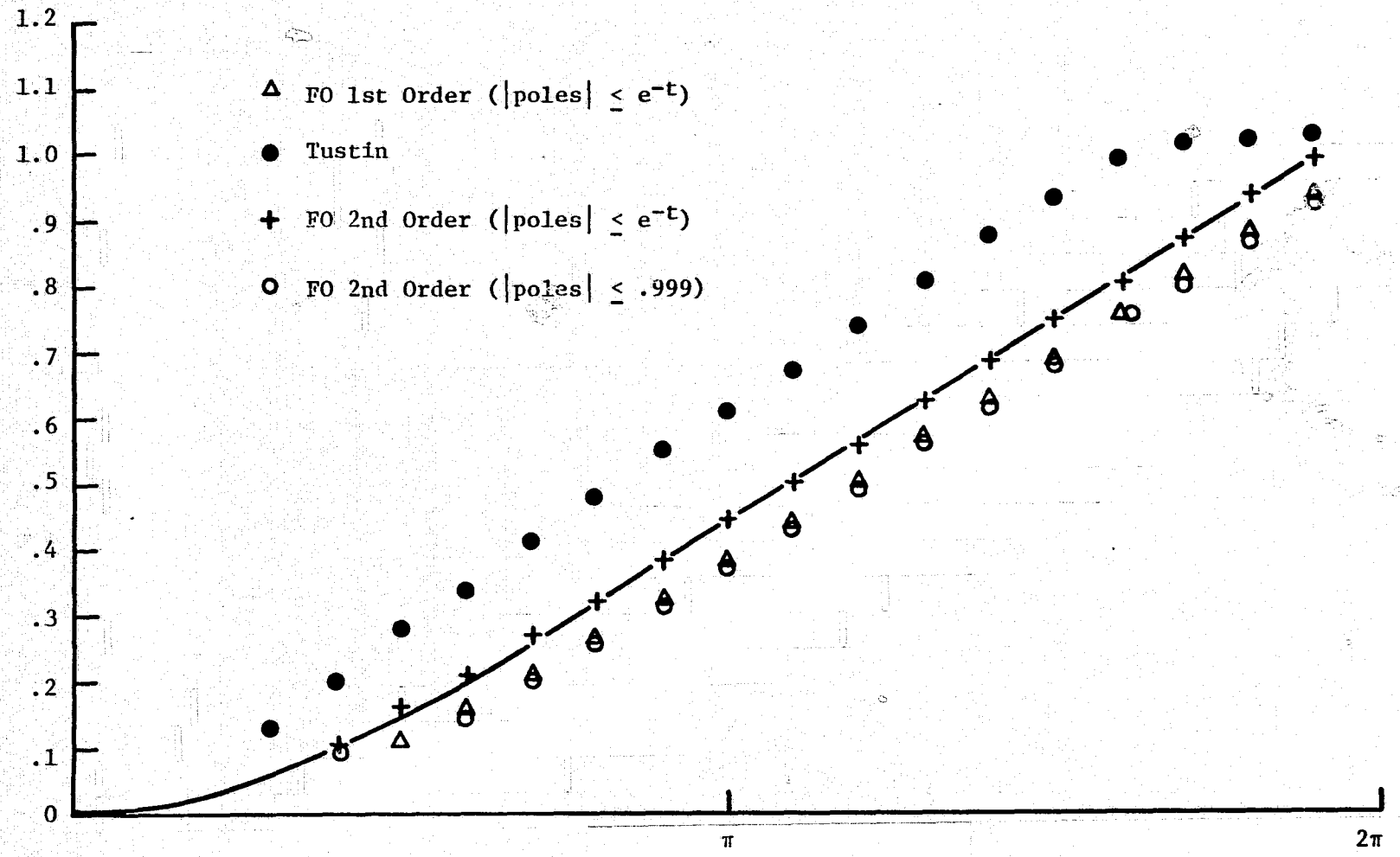


Figure 1-4 Comparison of Exact, Tustin, and FO Responses Ramp Input

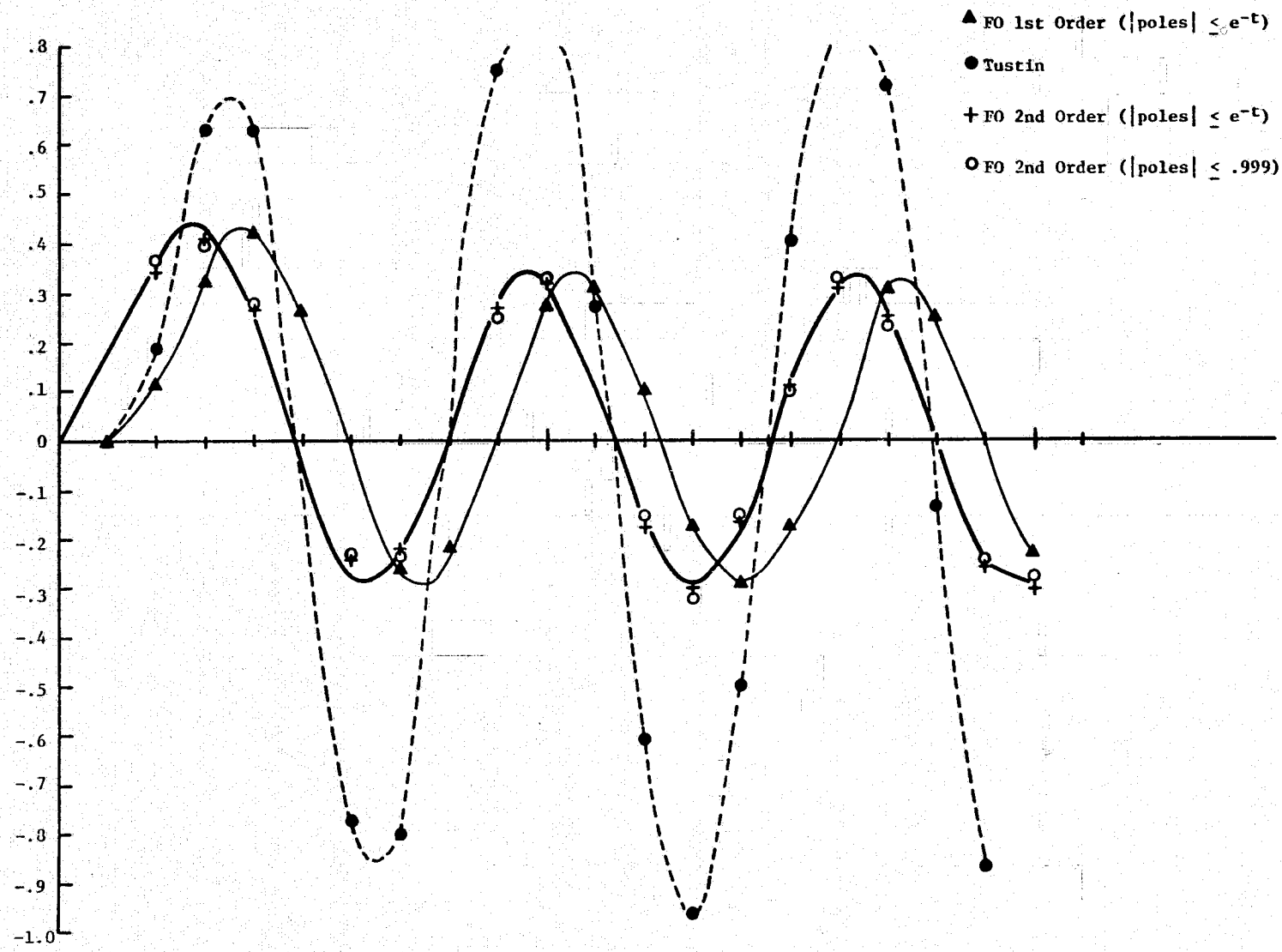


Figure 1-5 Comparison of Exact, Tustin, and FO Responses Sinusoidal Input

performance is best at smaller sampling periods. As we might expect, relaxing the pole magnitude constraint results in slight oscillations in the simulated responses. Empirical results also show that better time-domain accuracy can be obtained by using a digital system of one order higher than the simulated continuous system. This may result in slightly more computation time, but it is not particularly critical. The digital system can be designed with high sampling periods to accomodate the real-time constraint.

One disadvantage of the FO method is that it is not very flexible in terms of sampling period. Once a digital system has been obtained, the sampling period cannot be changed without degrading the simulation performance. If the designer desires a model for a different sampling period, he has to start the design process again. This characteristic is shown in Fig. 1-6, where the digital system had been designed for $T = \Pi/10$ secs.; but the simulation was performed at $T = .1$ sec. For sampling periods other than the one that has been designed for, the time-domain error increases substantially. Fortunately, this drawback is not particularly severe. The design procedure depends almost completely on the digital computer, and there is little calculation to be performed by the designer.

All in all, the FO method seems very promising from preliminary results obtained to date. Only first-order systems have been tried, and no conclusive results can be derived from this limited empirical data. The next task will include higher-order systems with a variety of inputs.

1.1.2 Time Domain Optimization

The preceding discussion was centered around the time-domain response resulting from a frequency domain optimization. Although there is a unique relationship between time and frequency response, minimizing frequency-domain

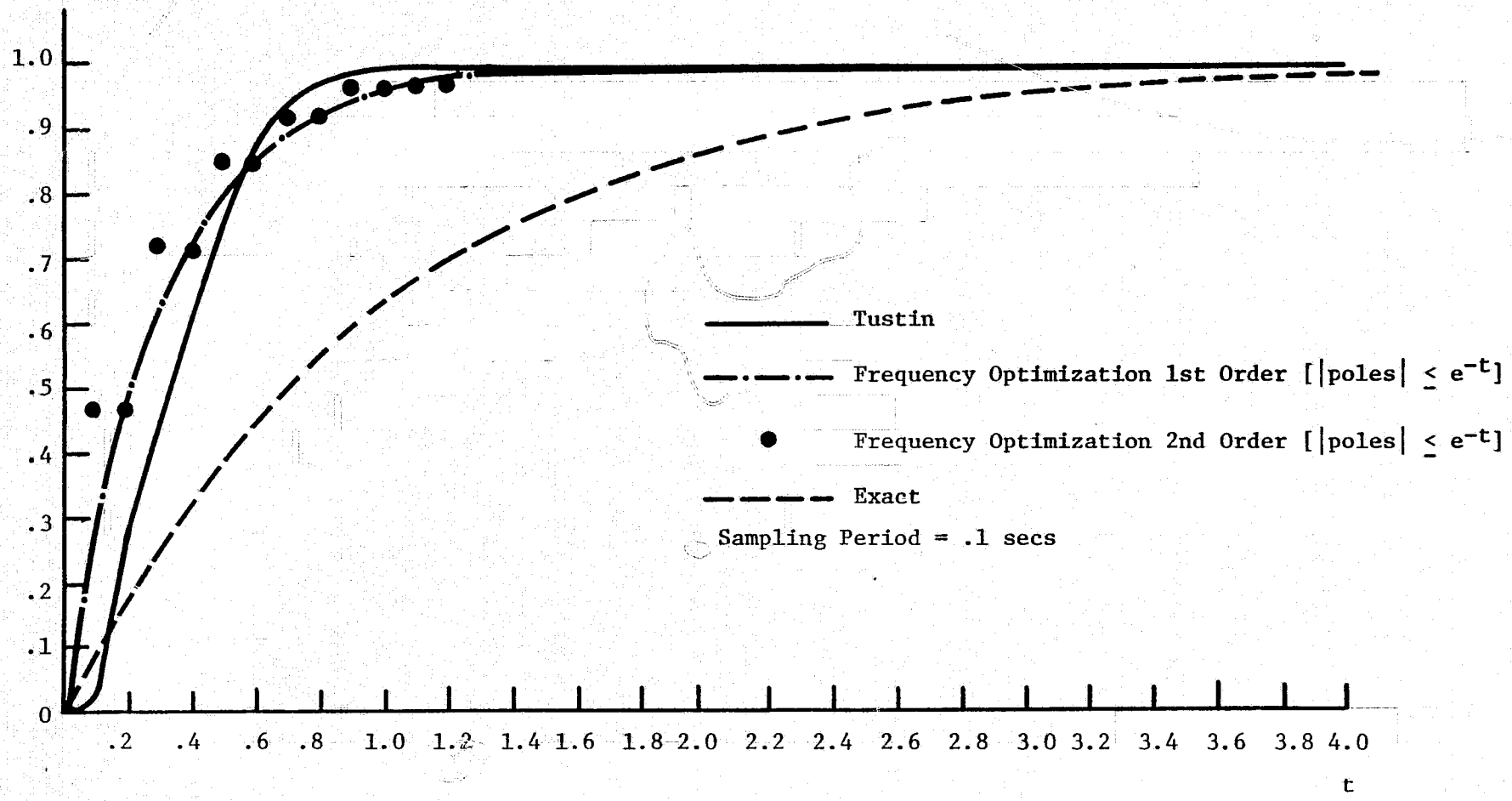


Figure 1-6 Comparison of Exact, FO, and Tustin Responses for T = .1sec

error does not necessarily make time-domain error a minimum. Our ultimate goal is to find an optimal time-domain simulation.

The state-of-the-art of optimal time-domain simulation is virtually non-existent. The biggest obstacle seems to be the definition of a general performance index whose value must be minimized to yield a desired digital system. Unlike the frequency domain performance index, the time-domain error is input dependent. This feature makes it difficult, if not impossible, to obtain a general performance index that will suit all inputs which vary from application to application. Presented below are some of the results published in the literature. Most of the methods presented are far from being applicable to our purposes. These approaches, however, could be used for our application, since the performance index depends only on the numerical values at the sampling instant of the ideal response.

J. A. Athanassopoulos and A. D. Warren [3] proposed the following form of the digital system;

$$H(\vec{\alpha}, z^{-1}) = \frac{\sum_{i=0}^{m} \alpha_{1i} z^{-1}}{\sum_{i=0}^{n} \alpha_{2i} z^{-1}} \tag{1-13}$$

where $$\vec{\alpha} = [\alpha_{10}, \alpha_{11}, \cdots, \alpha_{1m}, \alpha_{20}, \alpha_{21}, \cdots, \alpha_{2n}]^T \tag{1-14}$$

The time-domain response of the above system is a function of the coefficient vector α whose determination depends on the performance index.

Now, let r_k be a specific value for the response at $t = t_k (k = 1, 2, \cdots)$, let s be a set of all real α such that poles of $H(\vec{\alpha}, z^{-1})$ lie outside the unit circle, and let $R(\vec{\alpha}; k)$ be the response of

interest. In this application $R(\vec{\alpha}, k)$ is chosen to be the time domain response $y(\vec{\alpha}, t_k) = Y_k$.

The design problem is then of the following min-max form:

Given the specification set $\{r_k | k = 1, \cdots\}$ and the maximum allowable degree of H (i.e., maximum numerator and denominator order), find the coefficient vector $\vec{\alpha} = \vec{\alpha}^*$ such that:

$$\max_k |R(\vec{\alpha}^*, k) - r_k| \leq \max_k |R(\vec{\alpha}, k) - r_k| \tag{1-15}$$

Notice that $\vec{\alpha}^* \varepsilon$ s in order that the digital system be stable.

The above problem is converted to the form of a mathematical program by using an additional variable η. Thus, in order to solve (1-15) a vector $\hat{\alpha} \equiv [\vec{\alpha}, \eta]^T$ which minimizes η subject to the constraints

$$\Psi_1(\hat{\alpha}, k) \equiv R(\vec{\alpha}, k) + \gamma_{1k}\eta - r_k \geq 0 \tag{1-16}$$

$$\Psi_2(\hat{\alpha}, k) \equiv -R(\vec{\alpha}, k) + \gamma_{2k}\eta + r_k \geq 0 \tag{1-17}$$

must be determined. This can be easily seen in Fig. 1-7.

The goal is to minimize the width $(\gamma_{1k} + \gamma_{2k})\eta$ of a zone subject to the constraint that both $R(\vec{\alpha}, k)$ and r_k lie in the zone for all k. Various methods, including a penalty function, can be used if the constraints are highly nonlinear functions of the variables.

More studies must be directed to this method before it can be directly applied to our application. The constraint is only that poles must be inside the unit circle. This is not enough to obtain a good representation of the continuous system, since the optimization procedure will tend to place poles at

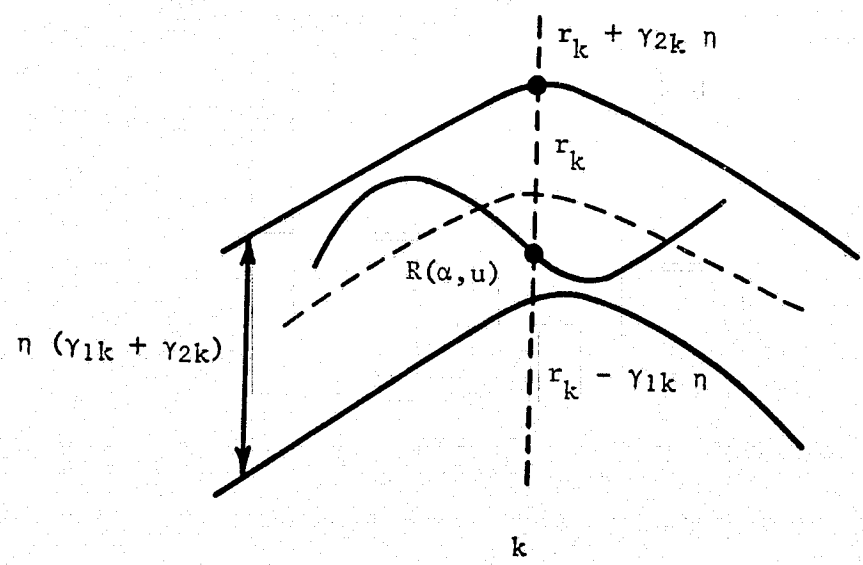


Figure 1-7 Optimization Problem

farthest possible location.

Other approaches proposed by Burrus and Parks [4], and Steiglitz [5] have appeared in the literature. Unfortunately, no stability considerations were given for these methods; and, as a result, they are not practical for our research. With these discussions in mind, we propose the following approach to our research in time-domain optimization.

The digital system has the same form as used in the FO method, i.e.,

$$\frac{Y(z)}{U(z)} = H(z) = A \frac{\prod_{k=1}^{K} (1 + a_{2k-1}z^{-1} + a_{2k}z^{-2})(1 + a^{1}_{NZ}z^{-1})}{\prod_{k=1}^{NCP/2} (1 + 2b_{2k}\cos b_{2k-1}z^{-1} + b^{2}_{2k}z^{-2}) \prod_{k=NCP+1}^{NP} (1 + b_{k}z^{-1})} \tag{1-18}$$

The advantages of this digital form were discussed in the semi-annual report.

The error function is defined in time domain as:

$$E = \sum_{n=0}^{N} |Y_n - Y(nT)| \tag{1-19}$$

where Y_n is the exact response and Y(nT) is the simulation response from (1-18) for a given input. This error function is input dependent; and, in order to achieve maximum generality, we must divide inputs into different classes and design the digital system accordingly. In other words, a prototype digital system is designed for each class of input; and the designer must choose between several systems for the best one under a given circumstance. This approach is an extension of the Sage's Optimum Discrete Approximation which only uses the step and ramp inputs as test inputs. We can extend this further and use a variety of inputs with the aid of the computer and the optimization

algorithm that is already available.

1.2 Digital Simulation and Optimization via Random Search Techniques

1.2.1 Introduction

This section continues discussion on our work involving techniques for the development of a discrete time integration operator to be used in the simulation process. The integration operator can be optimized for a particular system subjected to a set of specified inputs. The class of systems being investigated are those which can be represented by a set of state equations. A discrete time integration operator with certain free parameters is hypothesized. An adaptive random search optimization (ARSO) technique is used to find the optimum values for these parameters. Examples are presented to show the effectiveness of this technique.

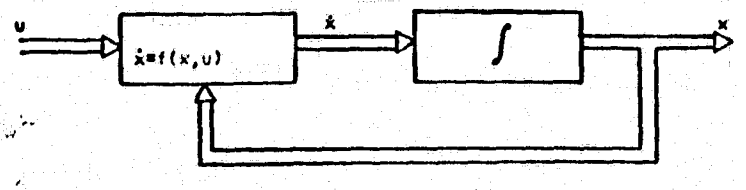
1.2.2 Integration Operator

The class of systems being investigated are those which can be represented by the set of state equations

$$\dot{x} = f(x, u) \tag{1-20}$$

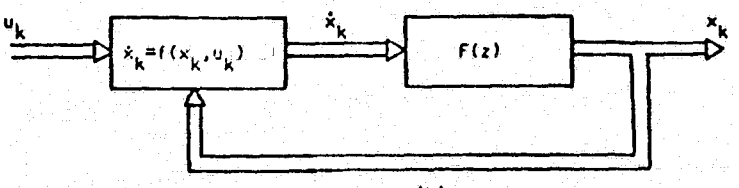
where x is the n x 1 state vector, u is for the r x 1 control vector, and f is the set of n functions, typically nonlinear.

Figure 1-8a is a block diagram of the mathematical relations in Eqn. (1-20). The vectors x and u are acted upon by the functional relations f(x, u), producing the vector $\dot{x}$. Figure 1-8b is a block diagram of a discrete approximation to the continuous time system. The control vector u is assumed to be sampled at a uniform rate, producing the input samples u(k). The equations



(a)

CONTINUOUS TIME SYSTEM



(b)

DISCRETE TIME SIMULATION

$$F(z) = \frac{\sum_{l=0}^{M} a_l z^l}{\sum_{j=0}^{N} b_j z^j}$$

N>M FOR REALIZABLE SIMULATION

Figure 1-8 Digital Simulation of Nonlinear Systems

$$\dot{x}(k) = f[x(k), u(k)] \tag{1-21}$$

are in the same form as those representing the continuous time system. A discrete-time integration operator of the following form is chosen.

$$F(z) = \frac{T \sum_{j=0}^{N} \lambda_j z^j}{z^N(z-1)} = \frac{T(\lambda_0 + \lambda_1 z + \cdots + \lambda_N z^N)}{z^{N+1} - z^N}$$

$$F(z^{-1}) = \frac{T[\lambda_N z^{-1} + \lambda_{N-1} z^{-2} + \cdots + \lambda_1 z^{-N} + \lambda_0 z^{-(N+1)}]}{1 - z^{-1}} \tag{1-22}$$

where T is the sampling period, and the λ's are a set of free parameters, the values of which are to be optimized. This operator yields a realizable simulation, since the power of the denominator is always one greater than that of the numerator. The pole at $z = 1$ corresponds to a pole at the origin in the complex s-plane, and the N^{th}-order pole at the origin in the z-plane corresponds to an N^{th}-order pole at negative infinity in the s-plane [6]. Therefore, the transient response of the poles added at $z = 0$ to make the operator closed-loop realizable will decay quickly. The state equations are now of the form

$$\dot{x}(k) = f[x(k), u(k)]$$

and

$$x(k+1) = x(k) + T[\lambda_N \dot{x}(k) + \lambda_{N-1} \dot{x}(k-1) + \cdots + \lambda_0 \dot{x}(k-N)] \tag{1-23}$$

Equation (1-23) can be thought of as a polynomial approximation to the value of a function at point $(k + 1)$ based on its value at point (k) and the value of its derivative at point (k) and preceding.

The free parameters in F(z) are optimized using an idealized model form of a model reference adaptive control system [7]. Figure 1-9 is a block diagram of this configuration.

1.2.3 Optimization Technique

The perturbation of parameters in F(z) is controlled by an Adaptive Random Search Optimization technique (ARSO) which was described in our previous report [1].

The unknown parameters are perturbed in the following manner:

$$\lambda_i(j + 1) = \lambda_i^* + \delta\lambda_i(j + 1) \tag{1-24}$$

where $\lambda(j + 1)$ is the new value of the i^{th} parameter, λ_i^* is the "best-to-date" value of the i^{th} parameter; that is, the value of λ_i when the minimum-to-date value of the J vector was calculated and $\delta\lambda_i(j + 1)$ is the random perturbation for the i^{th} parameter. This is equivalent to the perturbation scheme shown below

$$\lambda_i(j + 1) = \lambda_i(j) - a(j)\delta\lambda_i(j) + \delta\lambda_i(j + 1)$$

where

$$a(j) = 0, \text{ if } J_{(j)} < J^*_{(j - 1)}$$

$$= 1, \text{ if } J_{(j)} \geq J^*_{(j - 1)}$$

$J_{(j)}$ is the performance vector in the j^{th} trial, and $J^*_{(j - 1)}$ is the smallest performance vector obtained through $(j - 1)$ trials. The coefficient $a(j)$ is used to negate the effect of an unsuccessful trial. The perturbation is

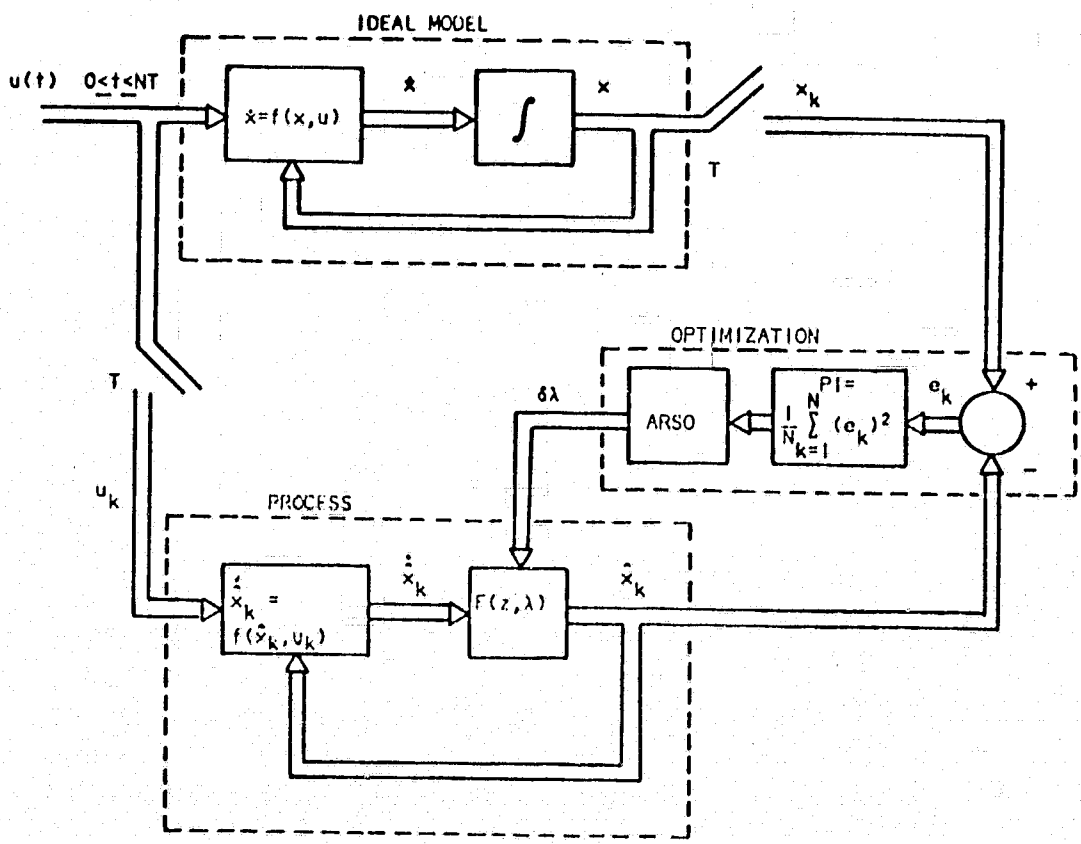


Figure 1-9 Optimal Digital Simulator Development Viewed as a Model Reference Adaptive Control System Using Adaptive Random Search Optimization Techniques

calculated as follows:

$$\delta\lambda_i(j) = \mu_i(j) + \sqrt{3\sigma_i^2(j)}[2RND(0) - 1] \qquad (1\text{-}25)$$

where $\mu_i(j)$ is the current value for the mean of the i^{th} random variable, $\sigma^2(j)$ is the current value for the variance, and RND(0) is a uniformly distributed random variable on the interval [0, 1]. Equation (1-25) produces a random number from a uniformly distributed random variable with mean μ and variance σ^2. Stability considerations may place constraints on the parameter calculated in (1-24). If a particular perturbation places a value outside its limit for stability, the value may be moved deterministically inside the limit, or another random perturbation may be tried. In simulating complex systems, however, it may be difficult to a priori determine the stability limits for the coefficients. In this case one or more of the state variables may be monitored during the simulation; and, if they exceed reasonable values, the trial may be aborted. This saves computation time and may prevent the entire program from being terminated due to overflow.

When a particular trial is successful, that is,

$$J_i \leq J_i^* \qquad 1 \leq i \leq n$$

where J_i^* is the minimum value of the i^{th} element of J, the means and variances of the random variables are updated. The mean is calculated as follows:

$$\mu_i(j) = \lambda^*_i - m_i \qquad (1\text{-}26)$$

where

$$m_i = \frac{c_i(1)}{2} + \frac{c_i(2)}{4} + \frac{c_i(3)}{8} + \frac{c_i(4)}{16} + \frac{c_i(5)}{16} \qquad (1\text{-}27)$$

The c_i's are past values of λ_i^*; that is, previous values of the "best-to-date" parameters. The most recent value of λ_i^* is $c_i(1)$. Since the most recent best value corresponds to a smaller index than a previous best value, (1-26) tends to select the most favorable direction for the next perturbation.

The variance for the distribution is determined using the following argument. The perturbation for each parameter is a random variable with a uniform probability density function with mean μ and variance σ^2. Relating these moments to the end points of the function (a, b) yields

$$\mu = (b + a)/2 \qquad (1\text{-}28a)$$

$$\sigma^2 = (b - a)^2/12 \qquad (1\text{-}28b)$$

If the mean and one end point is known, the other end point and variance can be calculated. That is, if μ and a are known, then

$$b = 2\mu - a \qquad (1\text{-}29a)$$

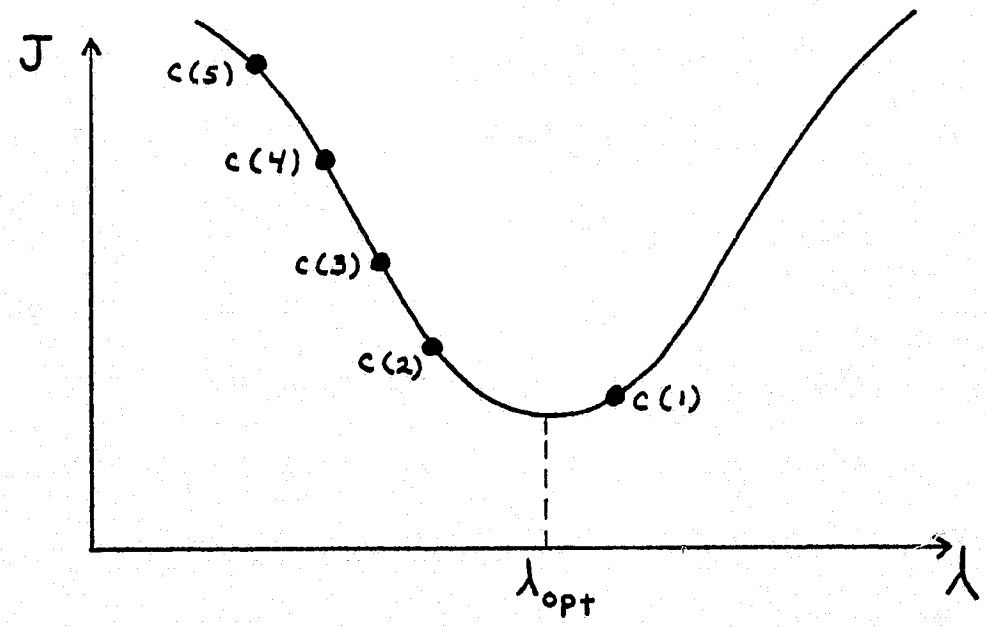
$$\sigma^2 = (\mu - a)^2/3 \qquad (1\text{-}29b)$$

and if μ and b are known, then

$$a = 2\mu - a \qquad (1\text{-}30a)$$

$$\sigma^2 = (b - \mu)^2/3 \qquad (1\text{-}30b)$$

Figure 1-10 illustrates two possible situations with a scalar cost function and a single parameter after five successes. In the top figure, c(2)



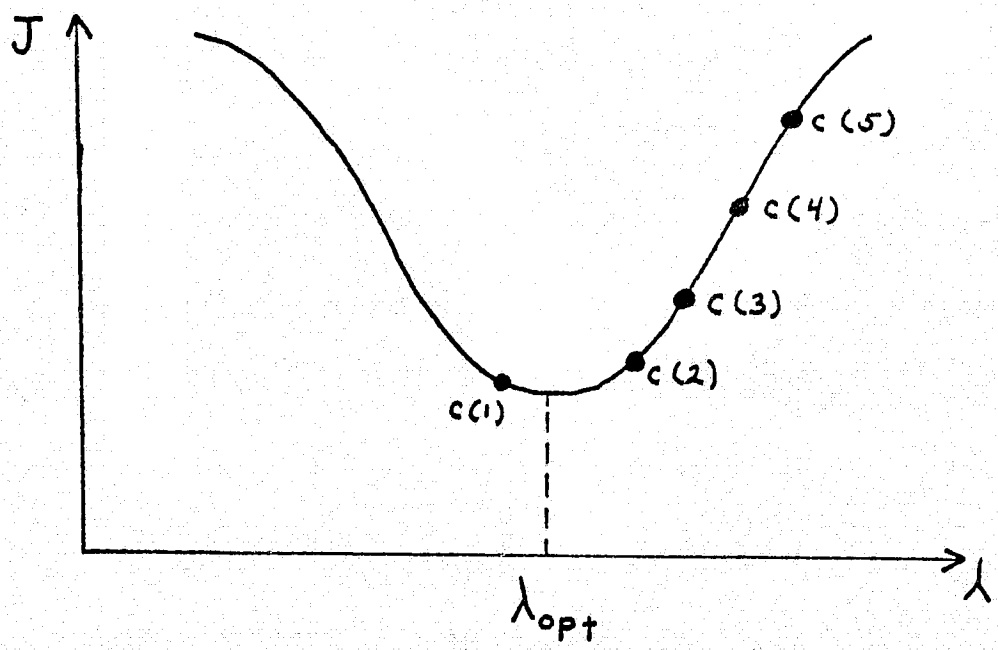


Figure 1-10 Sample Cost Functions

should be the end point (a) to ensure that the optimum parameter value lies within the search area. This gives

$$a = c(2) - c(1) \tag{1-31}$$

where c(1), the current best value, is taken as the zero point on the density function. Substituting this into (1-28) yields

$$\sigma^2 = [\mu - c(2) + c(1)]^2/3 \tag{1-32}$$

In the bottom figure c(2) should be the upper end point (b). Thus,

$$b = c(2) - c(1) \tag{1-33}$$

and

$$\sigma^2 = [c(2) - c(1) - \mu]^2/3 \tag{1-34}$$

Equations (1-32) and (1-34) yield the same numerical value for the variance, since the sign difference is lost when the numerator term is squared. Therefore, (1-34), combined with the mean from (1-26), provides the necessary data for computing the perturbations. When a large number of consecutive failures are generated, the means and variances are set to deterministic values. This allows the search technique to more fully explore the parameter space, about which little is known beforehand. If no improvements are obtained after a set number of failures, the search is terminated or restarted with a different set of initial conditions.

1.2.4 Results

One of the problems considered is the simulation of the nonlinear equations of an aircraft. These equations are shown below:

$$\dot{x} = V \cos(\gamma)$$

$$\dot{v} = -g \sin(\gamma) - \frac{d}{M} v^2 + \frac{T}{m} \cos(\alpha)$$

$$\dot{\gamma} = (1/v)(-g \cos(\gamma) + \frac{\ell}{M} v^2 + \frac{T}{m} \sin(\alpha)) \qquad (1\text{-}35)$$

$$\dot{\alpha} = \omega$$

$$\dot{\omega} = 1.311\ u - .806\ \omega - 1.311\ \alpha$$

where x is the horizontal displacement, h the vertical displacement, v the total velocity, γ the flight path angle, α the angle of attack, u the control elevator deflection, and ω the rate of change of the angle of attack.

Each of the six states generates an element of the performance index. The mean squared error between the approximate value and a value obtained by Runge-Kutta integration is used. The elements are:

J1 $\triangleq$ range error(feet)

J2 $\triangleq$ altitude error(feet)

J3 $\triangleq$ velocity error(feet/sec)

J4 $\triangleq$ error in flight path angle(radians)

J5 $\triangleq$ error in angle of attack(radians)

J6 $\triangleq$ error in angle of attack rate(rad/sec)

The RMS results of ARSO optimization using one, two, and three parameters are shown in Table 1-1. The sample period for each of these is 0.5 seconds. Also shown in the table are the results using the forward difference operator and the Milne-Reynolds predictor-corrector method. The Milne-Reynolds results are for a sampling period of 0.25 second , as both it and the delayed Tustin are unstable for T = 0.5 second. The total amount of computation per second

of simulation time was four times greater for Milne-Reynolds than for the discrete operators obtained via ARSO.

Table 1.1

Preliminary ARSO Results

	ARSO-1	ARSO-2	ARSO-3
J1	218	6.796	2.121
J2	4.39	3.63	3.385
J3	.305	.216	.205
J4	$8.76.10^{-4}$	$4.31.10^{-4}$	$4.16.10^{-4}$
J5	$3.0.10^{-2}$	$1.39.10^{-2}$	$1.37.10^{-2}$
J6	$3.4.10^{-2}$	$1.59.10^{-2}$	$1.57.10^{-2}$

	Fwd. Diff.	Milne Reynolds
J1	3.69	5.46×10^{-3}
J2	6.09	1.81×10^{-2}
J3	.377	3.22×10^{-5}
J4	7.4×10^{-4}	4.6×10^{-11}
J5	3.1×10^{-2}	2.7×10^{-8}
J6	3.5×10^{-2}	1.4×10^{-7}

ARSO-1: $\lambda_1 = 1.0127$

ARSO-2: $\lambda_1 = 1.20793$; $\lambda_2 = -.208282$

ARSO-3: $\lambda_1 = 1.20999$; $\lambda_2 = -.200159$; $\lambda_3 = -9.83883.10^{-3}$

The aircraft maneuver for the data in Table 1.1 was a climb followed by a

dive with a total range of approximately 5.6 miles. To evaluate the sensitivity of the cost function to changes in input, a second maneuver, a dive, was run. The total range was about the same as the first run. The results of this maneuver for ARSO-2 and ARSO-3 are shown in Table 1.2.

Table 1.2

	ARSO-2	ARSO-3
J1	8.96	2.82
J2	2.48	2.38
J3	.117	.113
J4	$4.13.10^{-4}$	$3.91.10^{-4}$
J5	$9.49.10^{-3}$	$9.38.10^{-3}$
J6	$1.1.10^{-2}$	$1,09.10^{-2}$

Using the ARSO-2 and ARSO-3 coefficients from maneuver one as initial conditions, the parameters were then optimized for input two. Table 1.3 presents the results of this optimization, and Table 1.4 lists the cost functions when maneuver one is run with the coefficients optimized for maneuver two.

Table 1.3

	ARSO-2	ARSO-3
J1	1.32	.653
J2	2.48	2.35
J3	.117	.111
J4	$4.04.10^{-4}$	$3.91.10^{-4}$
J5	$9.51.10^{-3}$	$9.3.10^{-3}$
J6	$1.1.10^{-2}$	$1.08.10^{-2}$

Table 1.4

	ARSO-2	ARSO-3
J1	3.99	2.92
J2	3.53	3.32
J3	.212	.200
J4	$4.34.10^{-4}$	$4.11.10^{-4}$
J5	$1.39.10^{-2}$	$1.35.10^{-2}$
J6	$1.60.10^{-2}$	$1.55.10^{-2}$

ARSO-2: $\lambda_1 = 1.20807$; $\lambda_2 = -0.207852$

ARSO-3: $\lambda_1 = 1.21322$; $\lambda_2 = -.201530$; $\lambda_3 = -1.15446.10^{-2}$

As can be seen from the tables, increasing the number of parameters decreases each element of the cost function in each case. This is because the higher-order polynomial approximation of (1-24) can better represent the actual function. It should be noted that, although the coefficients are input dependent, neither they nor the cost function change significantly when different maneuvers are executed. As an aid in evaluating the results of Tables 1.1-1.4, Table 1.5 lists the cost function for ARSO-2 from Table 1.1 as a percentage of the dynamic range for the respective state variables.

Table 1.5

	ARSO-2	Dynamic Range	Percent
J1	6.796	29,772	0.023
J2	3.63	304	1.19
J3	.216	14.318	1.51
J4	$4.31.10^{-4}$	.0525	0.82
J5	$1.39.10^{-2}$	1.0813	1.29
J6	$1.59.10^{-2}$	.2607	6.1

Since the performance vector does not allow an increase in value for any of the elements, the ARSO technique is dependent on initial parameter values. Thus, several different sets may be used before the final parameter values are determined.

1.2.5 Further Study

It is desired to apply the ARSO technique to a twelth-order set of equations representing a six degree of freedom aircraft. This will be a significant step forward in determining the general applicability of this optimization technique.

1.3 Some Options Available SDP

The existing programs of the SDP allow the user to simulate a continuous system up to 10^{th} order using the Tustin and Optimum Discrete Approximation methods. Results are compared with the fourth-order Runge-Kutta method to determine the error. The Runge-Kutta method is applied at a smaller sampling period to achieve "ideal" response.

Various plots can now be obtained. For comparison purposes the time response for each method is plotted against the ideal response for a given sampling period or each method is plotted for different sampling periods. Regardless of the type of plot the user may desire, there always is an "ideal" plot from which he can observe how well a given simulation method performs.

At the end of each simulation run, the user can also obtain an error plot. This error is based on the deviation from the Runge-Kutta method and is plotted against each sampling period that the user has previously specified.

To illustrate all the options available, two linear continuous systems were simulated. In all cases part (a) of a figure illustrates man/machine interactions, while part (b) shows the graphical results. The first example is a seventh-order autopilot transfer function. Step and sinusoidal inputs result in plots in Fig. 1-11 and Fig. 1-12 with sampling period of $T = .15708$ (40 rad/sec). The second example is a fourth-order continuous system. Figure 1-13 shows plots of responses obtained from the Tustin, Sage,

Figure 1-11 Step Response Using T = .15708 Illustrating Tustin and Sage Methods for

$$F(S) = \frac{36}{S^2 + 8.45 + 36} \cdot \frac{S^2 + 2.725 + 2.31}{S^2 + 5.62\ S + 3.1} \cdot \frac{S + 1.65}{S + .62} \cdot \frac{S^2 + 7.255 + 81}{1.125S^2 + 13.33\ S + 81}$$

```
INPUT NUMERATOR AND DENOMINATOR ORDERS
5,7
NUMERATOR COEFFICIENTS IN ASCENDING ORDER
11822.9,17970.9,11923.2,3719.74,358.72,32.
DENOMINATOR COEFFICENTS IN ASCENDING ORDER
4951.82,19049.1,23363.8,10433.1,24.61.6,340.468,26.4889,1.
INPUT INITIAL AND FINAL TIMES
0,15.708
HOW MANY SAMPLE PERIODS?
1
SAMPLE PERIODS-GREATER THAN.314E-01
.15708
RESPONSE LISTED? YES=0,NO=1
1
RESPONSE PLOTTED? YES=0,NO=1
0
PERFORMANCE INDEX:0-MSE,1-MAE,2-MTAE
0
INPUT: 0-STEP, 1-RAMP, 2-SINE
0
AMPLITUDE
1


TUSTIN NUMERATOR COEFFICIENTS
-.724E+07
.1975E+08
-.158E+08
-.126E+08
.5067E+08
-.364E+08
-.268E+08
.2999E+08
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
-.861E+07
.8118E+08
-.365E+09
.1003E+10
-.178E+10
.2016E+10
-.132E+10
.3713E+09
.0000E+00
.0000E+00
.0000E+00


P.I.          SAMP. PER.

.4868E-03         .157E+00

SAGE NUMERATOR COEFFICIENTS
.0000E+00
.0000E+00
-.335E+06
.2262E+07
-.666E+07
.1024E+08
-.799E+07
.2493E+07
.0000E+00
.0000E+00
.0000E+00

SAGE DENOMINATOR COEFFICIENTS
-.424E+06
.4730E+07
-.230E+08
.6313E+08
-.105E+09
.1042E+09
-.576E+08
.1356E+08
.0000E+00
.0000E+00
.0000E+00


P.I.          SAMP. PER.

.1244E-02         .157E+00


TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0007
SIMUL SUSP


.
```

Figure 1-11(a)

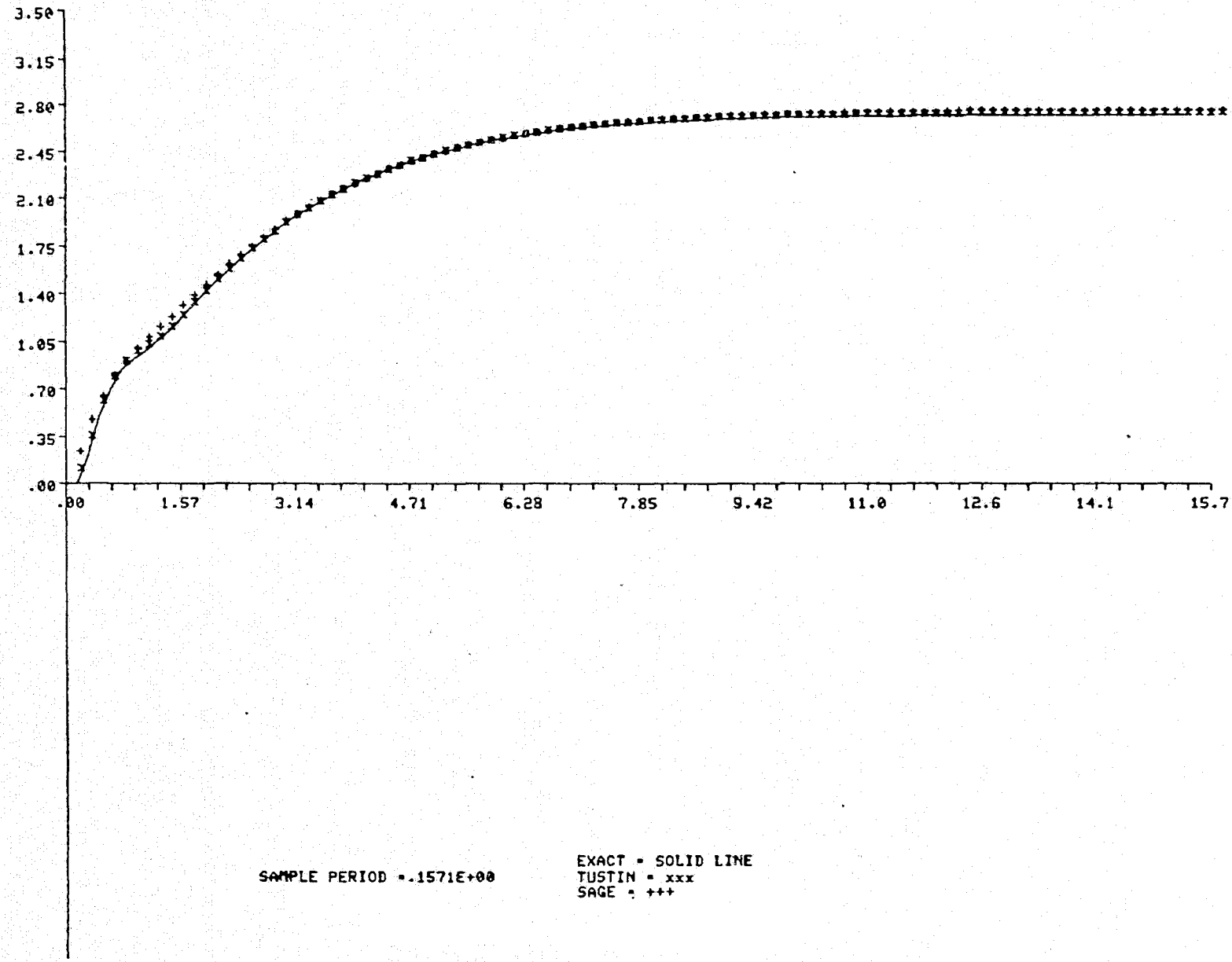

3.50
3.15
2.80
2.45
2.10
1.75
1.40
1.05
.70
.35
.00
.00
1.57
3.14
4.71
6.28
7.85
9.42
11.0
12.6
14.1
15.7
SAMPLE PERIOD = .1571E+00
EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++

Figure 1-11(b)

Figure 1-12 Sinusoidal Response (w = 1) for T = .15708
Illustrating Tustin and Sage Methods for

$$F(S) = \frac{36}{S^2 + 8.4S + 36} \cdot \frac{S^2 + 2.72\ S + 2.31}{S^2 + 5.62\ S + 3.1} \cdot \frac{S + 1.65}{S + .62} \cdot \frac{S^2 + 7.25\ S + 81}{1.125S^2 + 13.33 + 81}$$

```
INPUT NUMERATOR AND DENOMINATOR ORDERS
5,7
NUMERATOR COEFFICIENTS IN ASCENDING ORDER
11632.9,17970.9,11923.2,3719.74,358.72,32.
DENOMINATOR COEFFICIENTS IN ASCENDING ORDER
4951.82,19049.1,23363.8,10433.1,2461.6,340.468,26.4889,1.
INPUT INITIAL AND FINAL TIMES
0,15.708
HOW MANY SAMPLE PERIODS?
1
SAMPLE PERIODS-GREATER THAN.314E-01
.15708
RESPONSE LISTED? YES=0,NO=1
1
RESPONSE PLOTTED? YES=0,NO=1
0
PERFORMANCE INDEX:0-MSE,1-MAE,2-MTAE
0
INPUT: 0-STEP, 1-RAMP, 2-SINE
2
AMPLITUDE AND FREQ.-RAD/SEC
1,1


TUSTIN NUMERATOR COEFFICIENTS
-.724E+07
.1975E+08
-.158E+08
-.126E+08
.5067E+08
-.364E+08
-.268E+08
.2999E+08
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
-.861E+07
.8118E+08
-.365E+09
.1003E+10
-.178E+10
.2016E+10
-.132E+10
.3713E+09
.0000E+00
.0000E+00
.0000E+00


P.I.          SAMP. PER.

.7834E-03         .157E+00


SAGE NUMERATOR COEFFICIENTS
.0000E+00
.0000E+00
-.335E+06
.2262E+07
-.666E+07
.1024E+08
-.799E+07
.2493E+07
.0000E+00
.0000E+00
.0000E+00

SAGE DENOMINATOR COEFFICIENTS
-.424E+06
.4730E+07
-.230E+08
.6313E+08
-.105E+09
.1042E+09
-.576E+08
.1356E+08
.0000E+00
.0000E+00
.0000E+00


P.I.          SAMP. PER.

.4469E-03         .157E+00


TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0007
SIMUL SUSP


@
```

Figure 1-12(a)

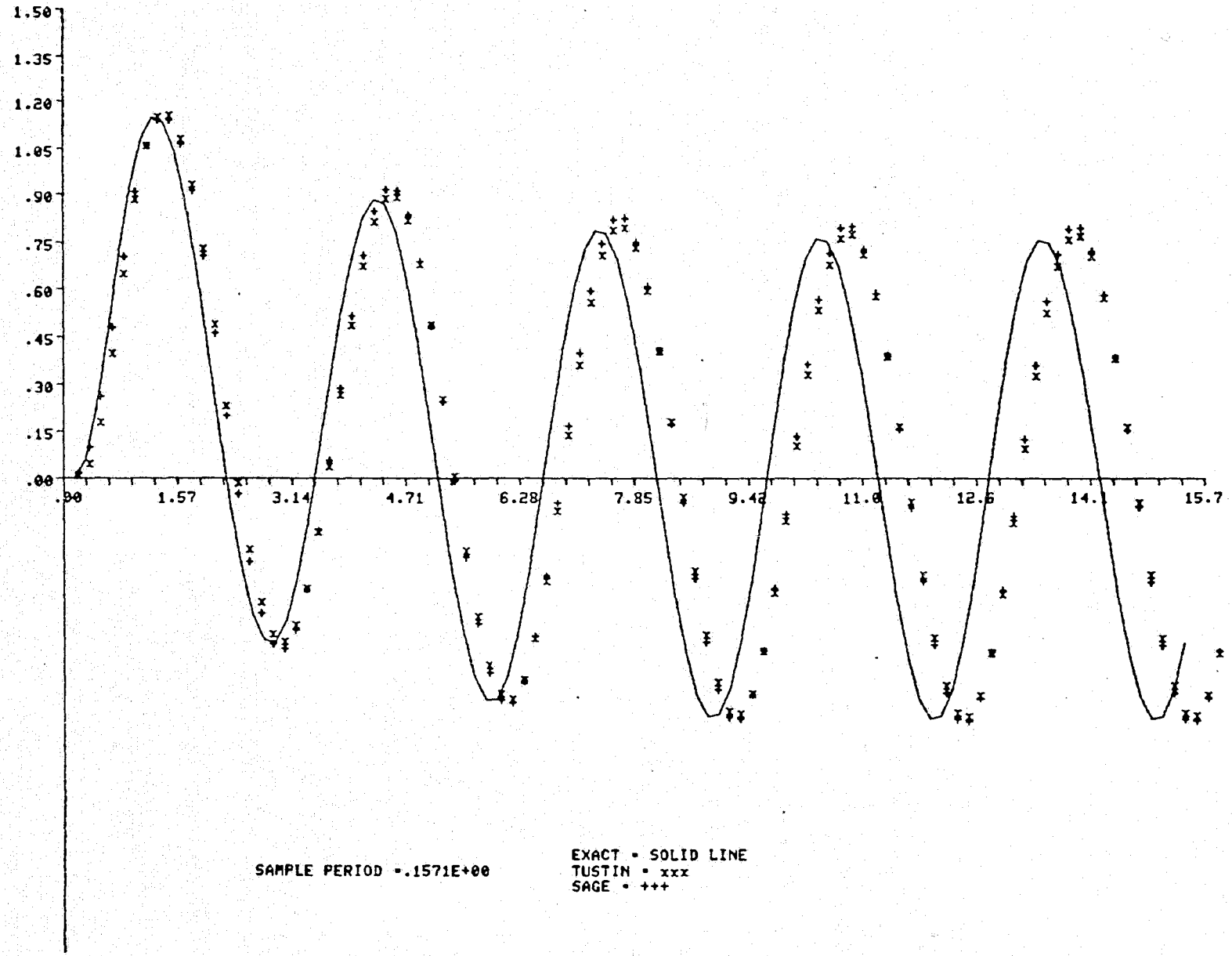

1.50
1.35
1.20
1.05
.90
.75
.60
.45
.30
.15
.00
.00
1.57
3.14
4.71
6.28
7.85
9.42
11.0
12.6
14.1
15.7
SAMPLE PERIOD = .1571E+00
EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++

Figure 1-12(b)

Figure 1-13 Step Response for T = .05 Illustrating Tustin and Sage Methods for

$$F(S) = \frac{S^2 + 5.5S + 2.5}{S^4 + 5S^3 + 9.5S^2 + 8S + 2.5}$$

```
INPUT NUMERATOR AND DENOMINATOR ORDERS
2,4
NUMERATOR COEFFICIENTS IN ASCENDING ORDER
2.5,5.5,1
DENOMINATOR COEFFICENTS IN ASCENDING ORDER
2.5,5,9.5,5,1
INPUT INITIAL AND FINAL TIMES
0,10
HOW MANY SAMPLE PERIODS?
1
SAMPLE PERIODS-GREATER THAN.200E-01
.05
RESPONSE LISTED? YES=0,NO=1
1
RESPONSE PLOTTED? YES=0,NO=1
0
PERFORMANCE INDEX:0-MSE,1-MAE,2-MTAE
0
INPUT: 0-STEP, 1-RAMP, 2-SINE
0
AMPLITUDE
1


TUSTIN NUMERATOR COEFFICIENTS
.1382E+04
-.430E+03
-.318E+04
.4500E+03
.1822E+04
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
.2255E+07
-.960E+07
.1533E+08
-.109E+08
.2896E+07
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00


P.I.          SAMP. PER.

.1262E-03         .500E-01


SAGE NUMERATOR COEFFICIENTS
.0000E+00
.0000E+00
.4000E+03
-.910E+03
.5125E+03
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

SAGE DENOMINATOR COEFFICIENTS
.1600E+06
-.680E+06
.1084E+07
-.768E+06
.2040E+06
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00


P.I.          SAMP. PER.

.1327E-03         .500E-01


TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0007
SIMUL SUSP


@
```

Figure 1-13(a)

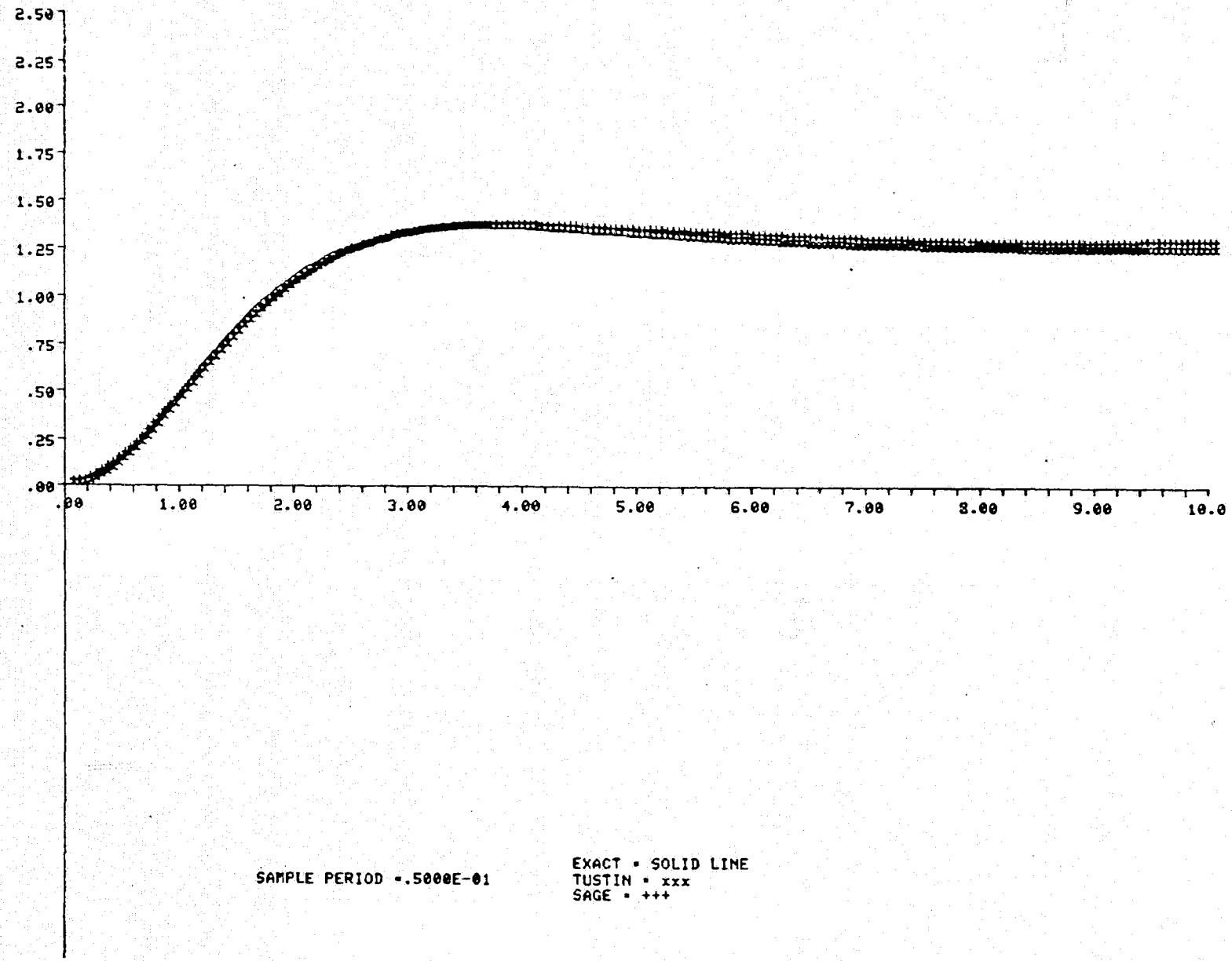

2.50
2.25
2.00
1.75
1.50
1.25
1.00
.75
.50
.25
.00
.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.0
SAMPLE PERIOD =.5000E-01
EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++

Figure 1-13(b)

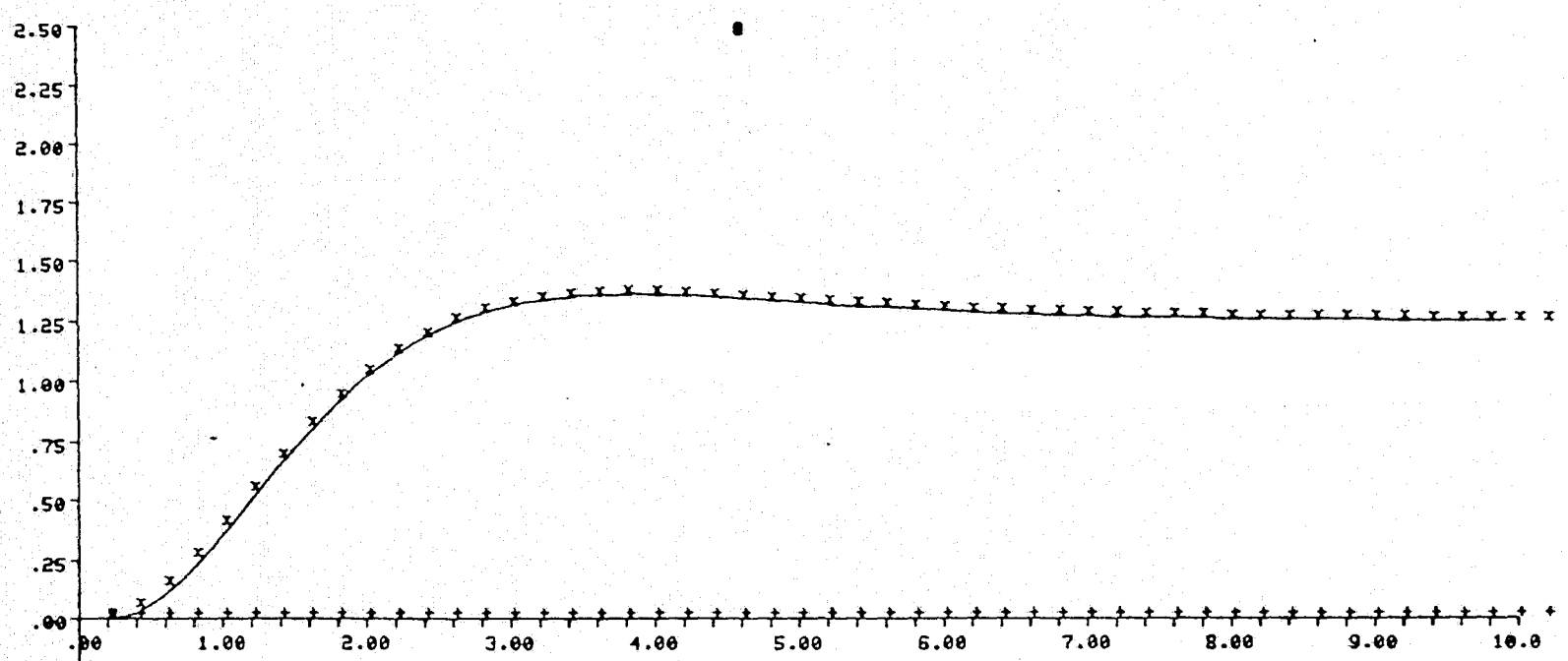
and fourth-order Runge-Kutta methods for a sampling period of T = .05 sec. Figures 1-14 through 1-18 are plots of responses of a given method (in this case, Tustin) with different sampling periods. The mean-square-error for each sampling period is calculated and is plotted in Fig. 1-19.

Figures 1-14 - 1-18 Step Response F or T = .2, .4, .6, .8 and 1.0, Respectively Illustrating Tustin Method for

$$F(S) = \frac{S^2 + 5.5\ S + 2.5}{S^4 + 5\ S^3 + 9.5S^2 + 8\ S + 2.5}$$

```
INPUT NUMERATOR AND DENOMINATOR ORDERS
2,4
NUMERATOR COEFFICIENTS IN ASCENDING ORDER
2.5,5.5,1
DENOMINATOR COEFFICENTS IN ASCENDING ORDER
2.5,6,9.5,5,1
INPUT INITIAL AND FINAL TIMES
0.,10.
HOW MANY SAMPLE PERIODS?
5
SAMPLE PERIODS-GREATER THAN.200E-01
.2,.4,.6,.8,1.
RESPONSE LISTED? YES-0,NO-1
1
RESPONSE PLOTTED? YES-0,NO-1
0
PERFORMANCE INDEX:0-MSE,1-MAE,2-MTAE
0
METHOD: 0-TUSTIN; 1-SAGE; 2-IBM
0
INPUT: 0-STEP, 1-RAMP, 2-SINE
0
AMPLITUDE
1


TUSTIN NUMERATOR COEFFICIENTS
.4750E+02
-1.00E+02
-.185E+03
.1200E+03
.1575E+03
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
.5872E+04
-.301E+05
.5811E+05
-.498E+05
.1603E+05
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00


TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0003
SIMUL SUSP

@
```

Figure 1-14(a)



SAMPLE PERIOD = .2000E+00

EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++

TYPE :GO TO CONTINUE
SIMUL : PAUSE 0004
SIMUL SUSP

Figure 1-14(b)

```
TUSTIN NUMERATOR COEFFICIENTS
.0000E+00
-.450E+02
-.350E+02
.6500E+02
.5500E+02
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
.2000E+03
-.132E+04
.3290E+04
-.366E+04
.1530E+04
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0003
SIMUL SUSP

•
```

Figure 1-15(a)

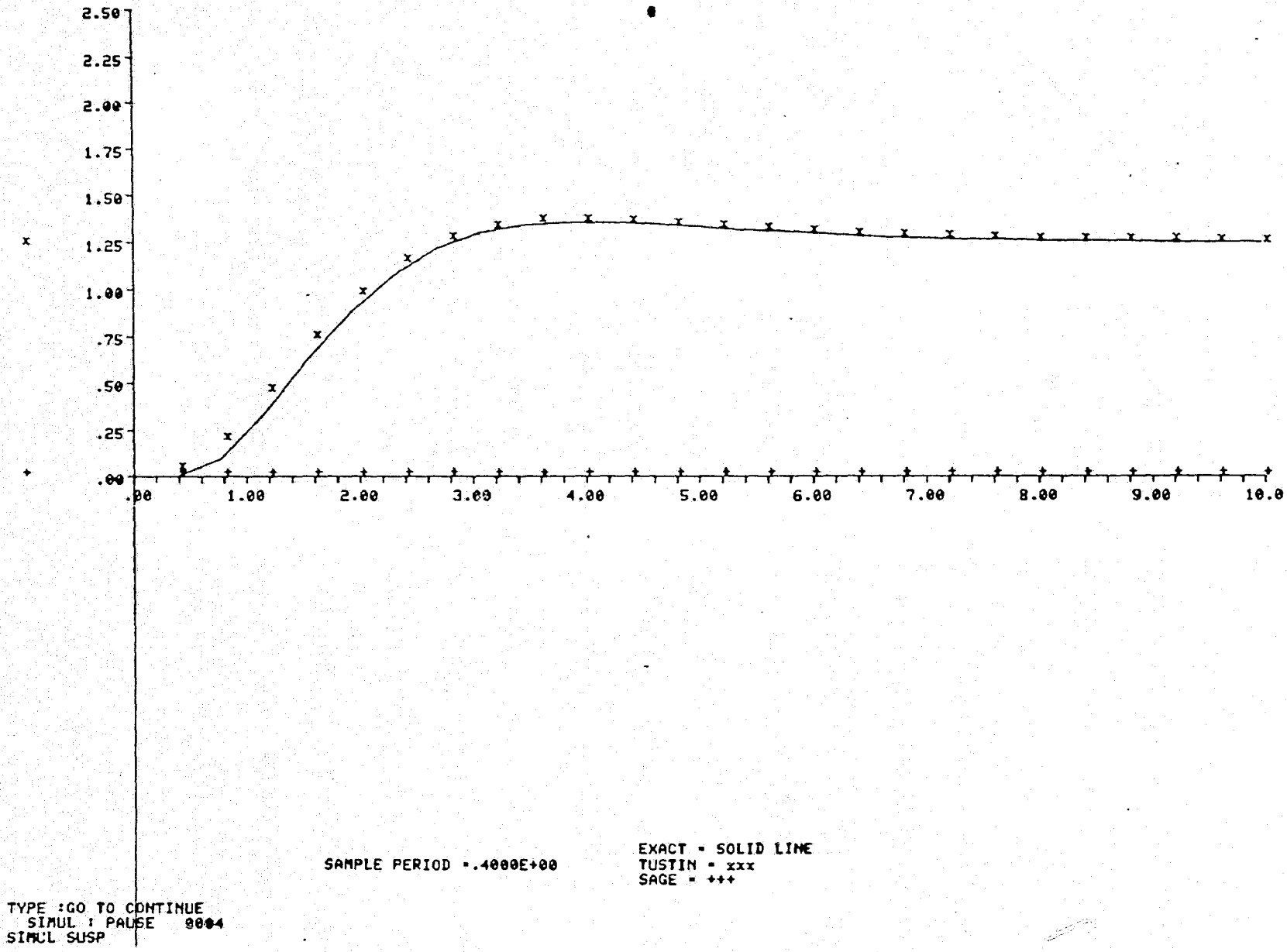

2.50
2.25
2.00
1.75
1.50
1.25
1.00
.75
.50
.25
.00
.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.0
SAMPLE PERIOD = .4000E+00
EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++
TYPE :GO TO CONTINUE
SIMUL : PAUSE 0004
SIMUL SUSP

Figure 1-15(b)

```
TUSTIN NUMERATOR COEFFICIENTS
-.472E+01
-.267E+02
-.722E+01
.4667E+02
.3194E+02
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
.1966E+02
-.167E+03
.5446E+03
-.801E+03
.4434E+03
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0003
SIMUL SUSP
```


Figure 1-16(a)

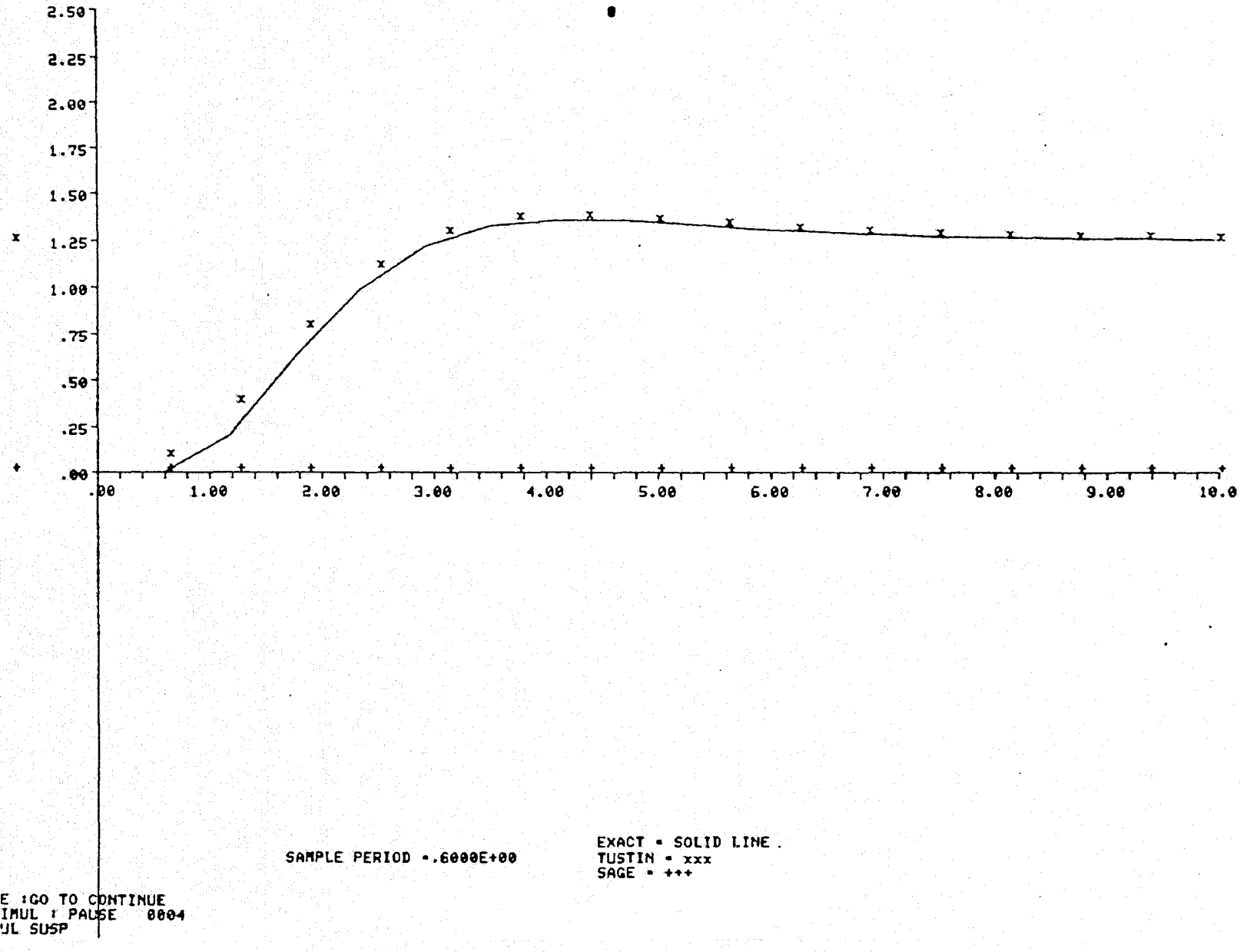

2.50
2.25
2.00
1.75
1.50
1.25
1.00
.75
.50
.25
.00
.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.0
SAMPLE PERIOD = .6000E+00
EXACT = SOLID LINE .
TUSTIN = xxx
SAGE = +++
TYPE :GO TO CONTINUE
SIMUL : PAUSE 0004
SIMUL SUSP

Figure 1-16(b)

```
TUSTIN NUMERATOR COEFFICIENTS
-.500E+01
-.175E+02
.2500E+01
.3750E+02
.2250E+02
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
.2812E+01
-.300E+02
.1306E+03
-.262E+03
.1991E+03
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00


TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0003
SIMUL SUSP
```


Figure 1-17(a)

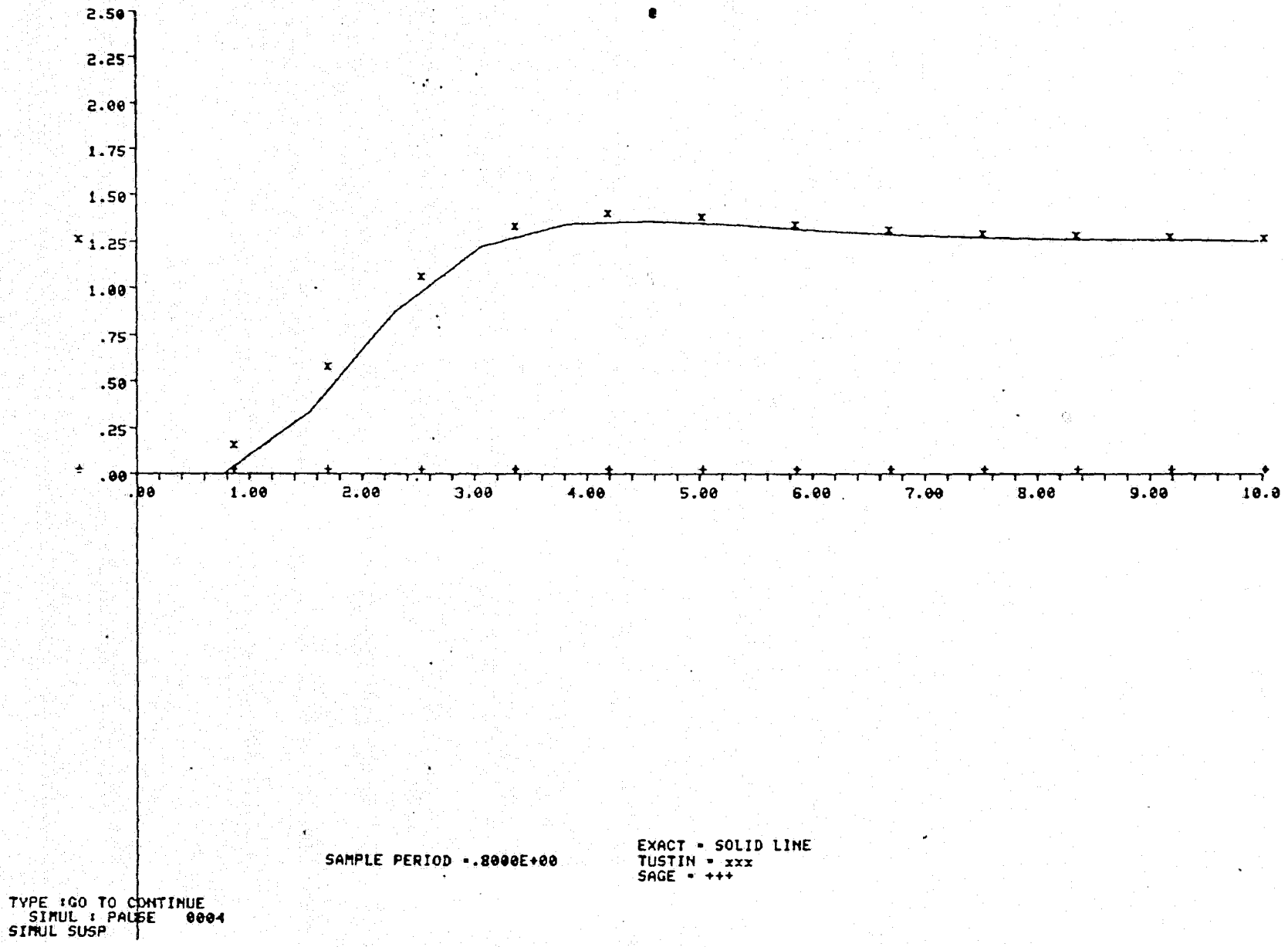
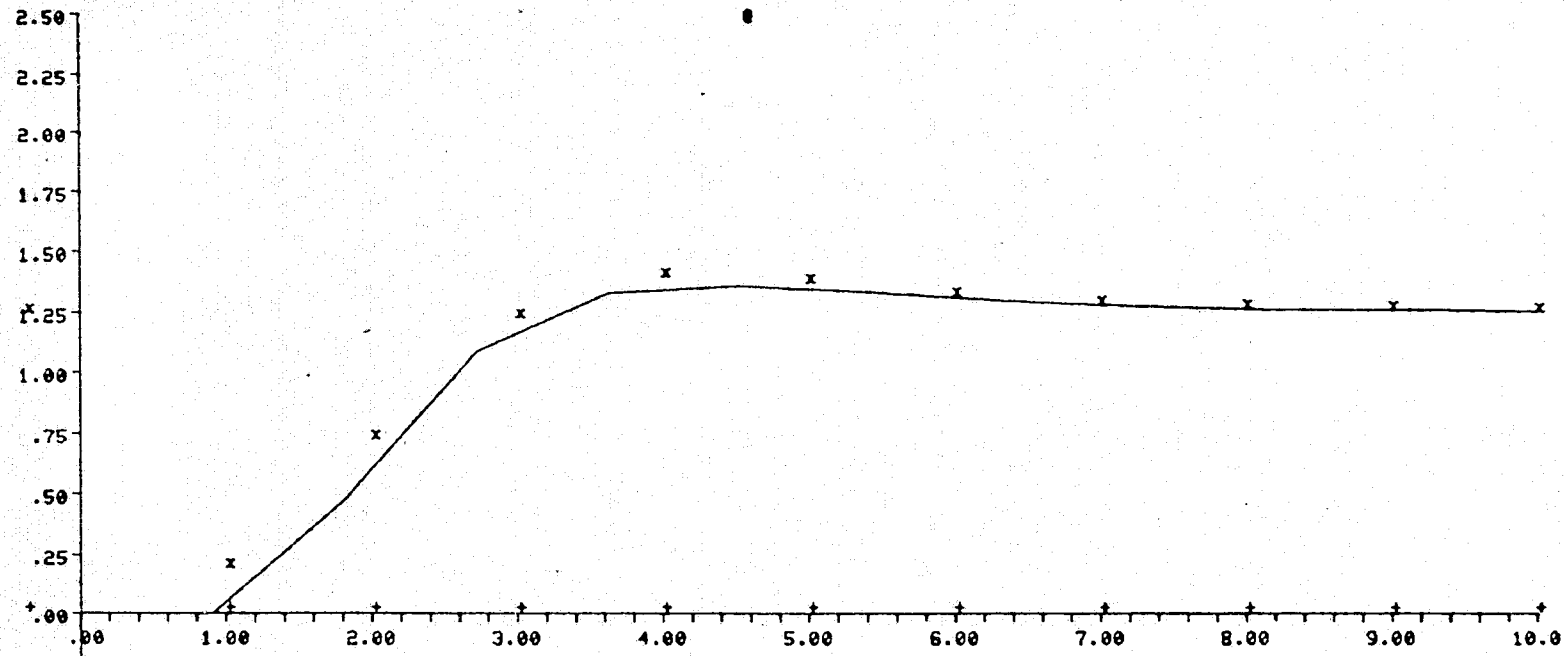

2.50
2.25
2.00
1.75
1.50
1.25
1.00
.75
.50
.25
.00
.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.0
SAMPLE PERIOD = .8000E+00
EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++
TYPE :GO TO CONTINUE
SIMUL : PAUSE 0004
SIMUL SUSP

Figure 1-17(b)

```
TUSTIN NUMERATOR COEFFICIENTS
-.450E+01
-.120E+02
.7000E+01
.3200E+02
.1750E+02
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TUSTIN DENOMINATOR COEFFICIENTS
.5000E+00
-.600E+01
.3500E+02
-.102E+03
.1125E+03
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00
.0000E+00

TYPE :GO FOR RESPONSE PLOT
  SIMUL : PAUSE   0003
SIMUL SUSP

•
```

Figure 1-18(a)



SAMPLE PERIOD = .1000E+01

EXACT = SOLID LINE
TUSTIN = xxx
SAGE = +++

TYPE :GO TO CONTINUE
SIMUL : PAUSE 0004
SIMUL SUSP

Figure 1-18(b)

Figure 1-19 Error Plot Corresponding to Cases in Figures 1-14 - 1-18.

```
P.I.          SAMP. PER.

.4226E-03          .200E+00
.1547E-02          .400E+00
.3304E-02          .600E+00
.5362E-02          .800E+00
.7403E-02          .100E+01

TYPE !GO FOR ERROR PLOT
  SIMUL : PAUSE    0001
SIMUL SUSP

@
```

Figure 1-19(a)

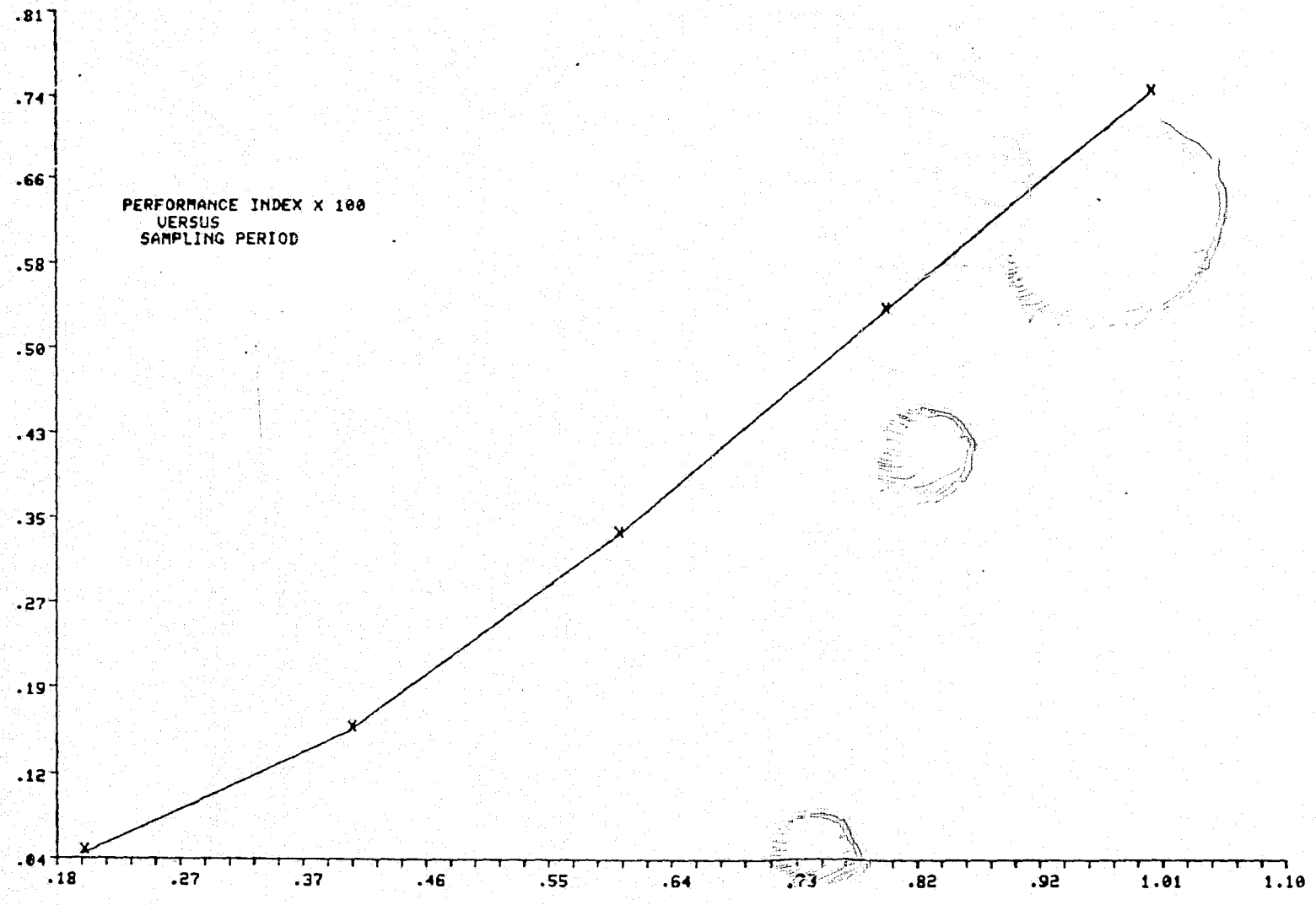

PERFORMANCE INDEX X 100
VERSUS
SAMPLING PERIOD
.81
.74
.66
.58
.50
.43
.35
.27
.19
.12
.04
.18
.27
.37
.46
.55
.64
.73
.82
.92
1.01
1.10

Figure 1-19(b)

II. PARALLEL OPERATION OF MICROPROCESSORS FOR REAL-TIME FLIGHT SIMULATION

2.0 INTRODUCTION

The use of parallel processing with microprocessors is being studied to determine if the new microprocessor technology can yield performance superior to that of large digital computers presently being used for real-time flight simulation. The simulation cycle time for large digital computers is typically about 1/32 second, and this sometimes introduces appreciable phase shift which leads to erroneous simulated results for dynamic systems. Hopefully, the use of several small processors would allow a decrease in cycle time; but the word length would almost certainly be reduced as a practical matter. Thus, one of the fundamental considerations in the study is to determine relationships between roundoff error and truncation error, i.e., the relationships between small word size with high speed and large word size and slow speed.

It is assumed, for the present, that high speed can be obtained via parallel processing; but this has not yet been established. The processing methodology and machine architecture to accomplish this are, themselves, major research problems under study.

As implied above, it is known that the round-off error can become more significant when the sampling rate (i.e., cycle time, extropolation time, etc.) becomes faster. Thus, it is important to determine the best range of sampling rates in terms of computational accuracy and stability as a function of truncation and round-off error. Once this range of sampling rates is determined, ways to implement the simulation on a multi-microprocessor system can be sought which will meet a satisfactory sampling rate specification.

Generally, the truncation error (and the total propagated error, as well) in simulation depends not only on the sampling rate and the simulation technique used but also on the characteristic frequency of the system to be

simulated. Study has been made in the literature [8] on the relative contribution of the truncation and the round-off error to the error committed at each step as a function of system characteristic frequency and the sampling rate, as well as the number of bits allotted to the mantissa. A brief review follows.

Let us assume that the j^{th} derivative of the state variables of the system are given by the relation

$$x^{(j)} = \omega_c{}^j x \tag{2-1}$$

where the characteristic frequency ω_c is, in general, complex. A Taylor series expansion of x from the point t = nh to t = (n + 1)h gives

$$x_{n+1} = x_n + h\dot{x}_n + \frac{1}{2} h^2 \ddot{x}_n + \frac{1}{6} h^3 \dddot{x}_n + \cdots \tag{2-2}$$

Substituting (2-1) into (2-2) yields

$$x_{n+1} = (1 + h\omega_c + \frac{1}{2} h^2 \omega_c{}^2 + \frac{1}{6} h^3 \omega_c{}^3 + \cdots)x_n \tag{2-3}$$

for which the solution using first-order integration is

$$x_{n+1} = (1 + h\omega_c)x_n \tag{2-4}$$

with the local truncation error

$$T_1 = \frac{1}{2} h^2 \omega_c{}^2 + O(h^3) \tag{2-5}$$

The change in x to first order relative to x is

$$\frac{1}{x_n} \Delta x_n = h\omega_c \tag{2-6}$$

The magnitude of the local relative truncation error is approximately

$$\varepsilon_{tr} \doteq \frac{|T_1|}{|\Delta x_n|} = \frac{1}{2}|h\omega_c| \tag{2-7}$$

where higher-order terms in (2-5) are neglected. Similar expressions can be derived for second-order integration, and they are

$$\frac{1}{x_n} \Delta x_n = h\omega_c(1 + \frac{1}{2} h\omega_c) \tag{2-8}$$

$$\varepsilon_{tr} = \frac{|T_2|}{|\Delta x_n|} = \frac{1}{6} \frac{|h^2 \omega_c{}^2|}{|1 + \frac{1}{2} h\omega_c|} \tag{2-9}$$

The local round-off error R is, when floating-point arithmetic is used, bounded by*

$$-2^{-N_b}\left|x_n\right| = R = 2^{-N_b}\left|x_n\right| \tag{2-10}$$

where N_b is the number of bits for the mantissa. The magnitude of the local round-off error ε_r is defined as

$$\varepsilon_r = \frac{|R|}{|\Delta x_n|} \tag{2-11}$$

---

*In this case it is assumed that the so-called inherent error can be neglected with good programming practices. For more detail refer to Ref. [8].

Thus, to a first-order approximation it has a maximum of

$$\varepsilon_r = \frac{2^{-N_b}}{|h\omega_c|} \tag{2-12}$$

and for second-order it has a maximum of

$$\varepsilon_r = \frac{2^{-N_b}}{|h\omega_c(1 + \frac{1}{2} h\omega_c)|} \tag{2-13}$$

It should be noted that the round-off error committed at each step is assumed to be random. For purpose of analysis a statistical model is generally adopted for the distribution of round-off error. The most common model is to assume that the round-off error values are uniformly distributed over the interval $(0, \varepsilon_r)$. For the assumed distribution, the 50 percentile occurs at the mid-point, namely $(1/2)\varepsilon_r$.

Figure 2-1 shows the local relative truncation and the round-off error for first-order and second-order integration expressed by (2-7) and (2-9), (2-12) and (2-13) as functions of $h\omega_c$. Two values are chosen for N_b, one typical of a large digital computer and the other a microcomputer with the maximum processing capability currently available*. The figure allows a convenient comparison of round-off and truncation errors. For example, when the second-order Adams-Bashforth (AB-2) method is used on a machine with $N_b = 47$ bits and

---

*Typical word length of a large digital computer is 60 bits. Out of 60 bits, 48 bits are assigned for mantissa. Since one bit is reserved for sign of the mantissa, the number of significant bits in the mantissa is $N_b = 47$. The maximum word length of currently available microprocessors in floating-point operation is 32 bits. Of the 32 bits the mantissa occupies 24 bits with one bit for sign, thus leaving 23 bits in the floating-point representation.

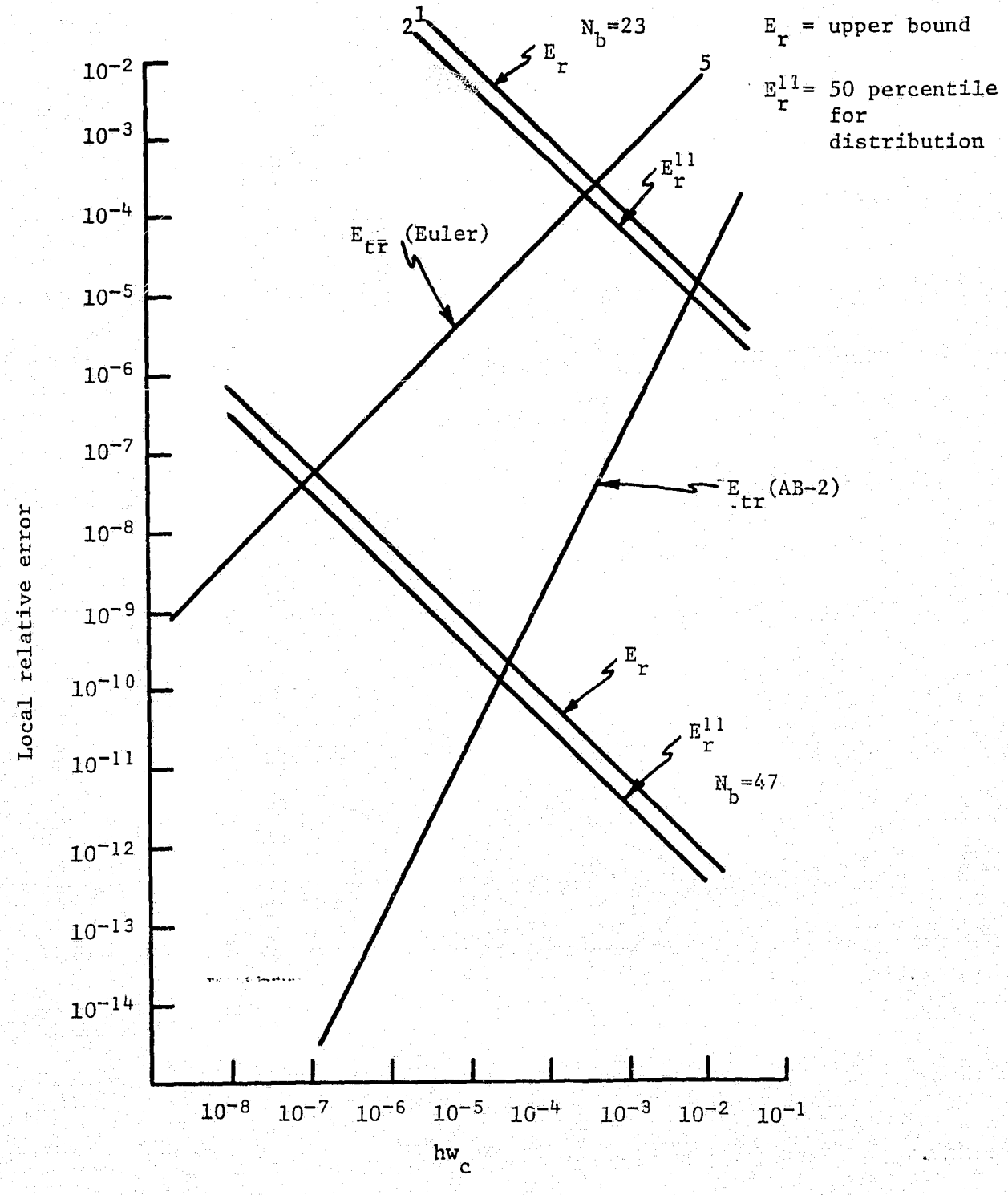


Figure 2-1 Local relative roundoff and truncation errors as a function of characteristic frequency and interval size product for a first- and second-order techniques on a 23-bit and a 47-bit fractional part machine.

$h\omega_c > 10^{-4}$ to 10^{-3}, the truncation error is at least two orders of magnitude larger than round-off error, while for $h\omega_c > 10^{-6}$ to 10^{-5} the situation is just the opposite.

When, at each local step, one type of error is much larger than the other in magnitude, it is considered that the propagated error is dominated by the former. Figure 2-1 will be referred to again in a later discussion.

A practical example has been considered*. The example system is described by

$$\dot{x} = Ax + Bu \tag{2-14}$$

where

$$A = \begin{bmatrix} -0.3236575 & 0 & 1 \\ 0 & 0 & 1 \\ -1.169521 & 0 & -0.4809339 \end{bmatrix}, \quad B = \begin{bmatrix} -0.01785196 \\ 0 \\ -1.379406 \end{bmatrix}$$

The characteristic roots for the above system are

$$\begin{aligned} S_1 &= 0 \\ S_2, S_3 &= -0.4023 \pm j1.0786 = 1.1512e^{j(180\pm69.54)^\circ} \end{aligned} \tag{2-15}$$

The step responses using the Euler and second-order Adams-Bashforth methods on a microprocessor with 32-bit floating-point arithmetic have been simulated by a Hewlett-Packard minicomputer which has 32-bit floating-point arithmetic. The system responses obtained have been compared with the solution from

---

*This is the short-period approximation for linearized longitudinal aircraft dynamic equations of motion, where $x_1 = \alpha$(angle of attack), $x_2 = \theta$(pitch angle), and $x_3 = \dot{\theta}$(pitch rate).

a fourth-order Runge-Kutta method using h = .001 on a computer with 60-bit floating-point arithmetic. This solution is considered to be relatively accurate. The average absolute relative error is shown in Figs. 2-2, 2-3, and 2-4 for each state variable, respectively, as a function of sampling frequency. It is interesting to note that the total error which is the combinational effect of the truncation and round-off error becomes minimum at a certain frequency for each state variable. Refer to Fig. 2-1 for an explanation of the results in Figs. 2-2, 2-3, and 2-4.

The magnitude of the characteristic roots for the example system is, from (2-15),

$$|\omega_c| = 1.1512 \tag{2-16}$$

From Figs. 2-3 and 2-5 the propagated error exhibits a sharp minimum at

$$h = 2 \text{ msec} \tag{2-17}$$

Thus,

$$|h\omega_c| \cong 0.0023 \tag{2-18}$$

From Fig. 2-1, loci 1 and 6 intersect at $h\omega_c \cong .01$, i.e., the truncation error with the second-order integration method and the maximum round-off error on the 23-bit machine committed at each step would be almost the same in magnitude as at $h\omega_c \cong .01$, thus competing with each other for contribution to the propagated error. Because the slopes of the two loci 1 and 6 are equal in magnitude but opposite in sign, we could reasonably have expected that the combinational effect would become minimum around the intersection point of line 1 and 6. It is noted that the value for $|h\omega_c|$ in (2-18) is farily close to, but

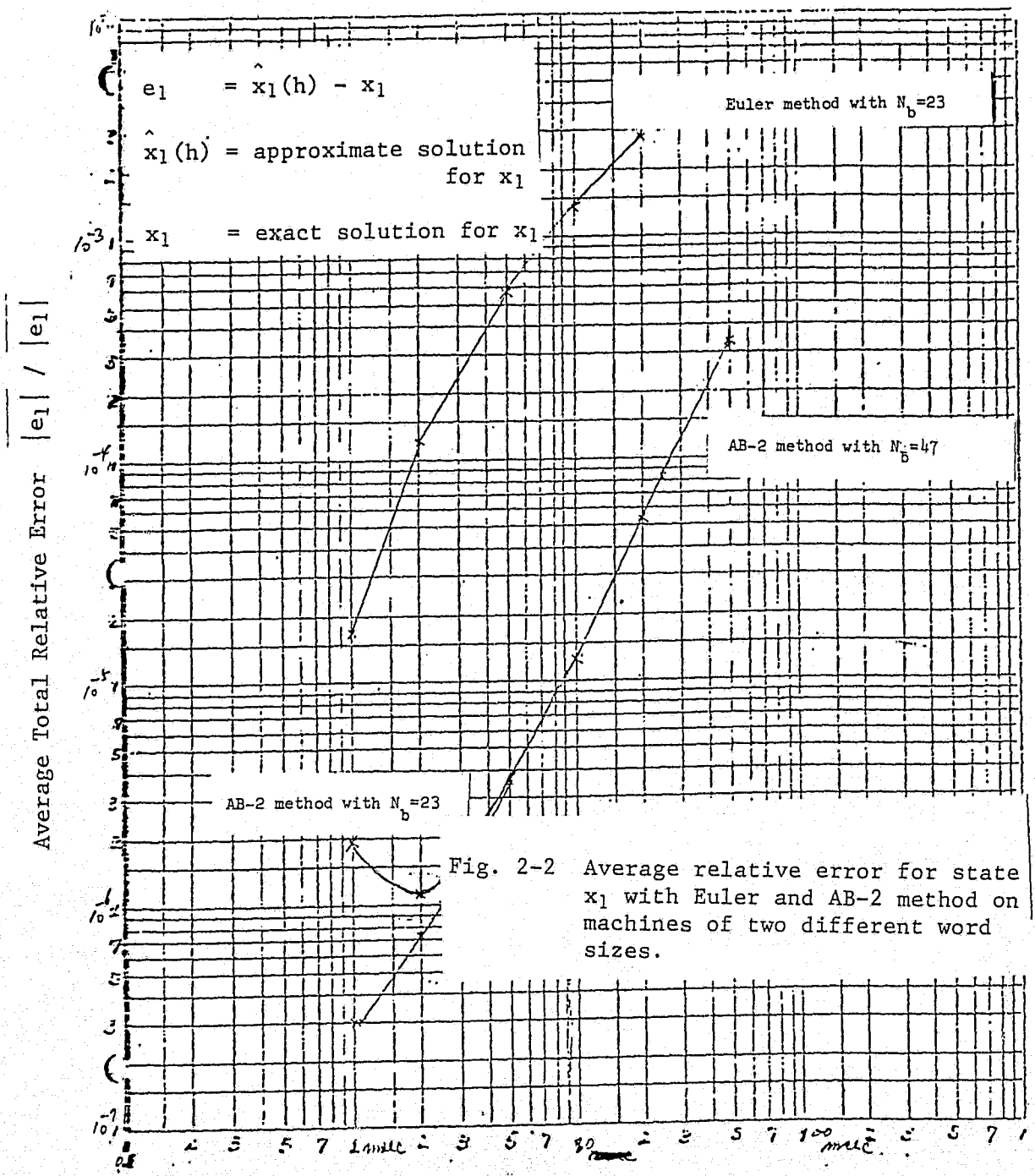


Fig. 2-2 Average relative error for state x_1 with Euler and AB-2 method on machines of two different word sizes.

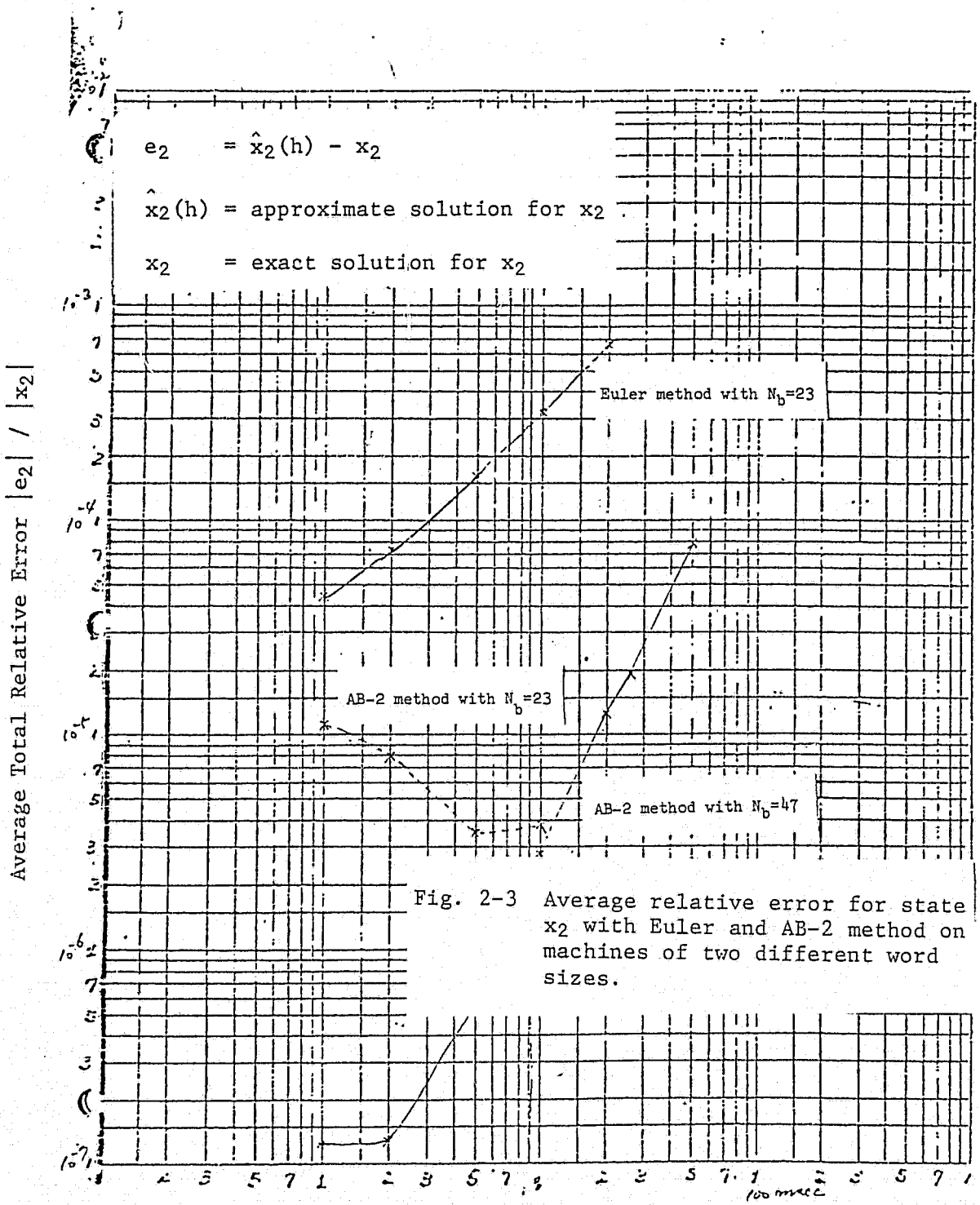


Fig. 2-3 Average relative error for state x_2 with Euler and AB-2 method on machines of two different word sizes.

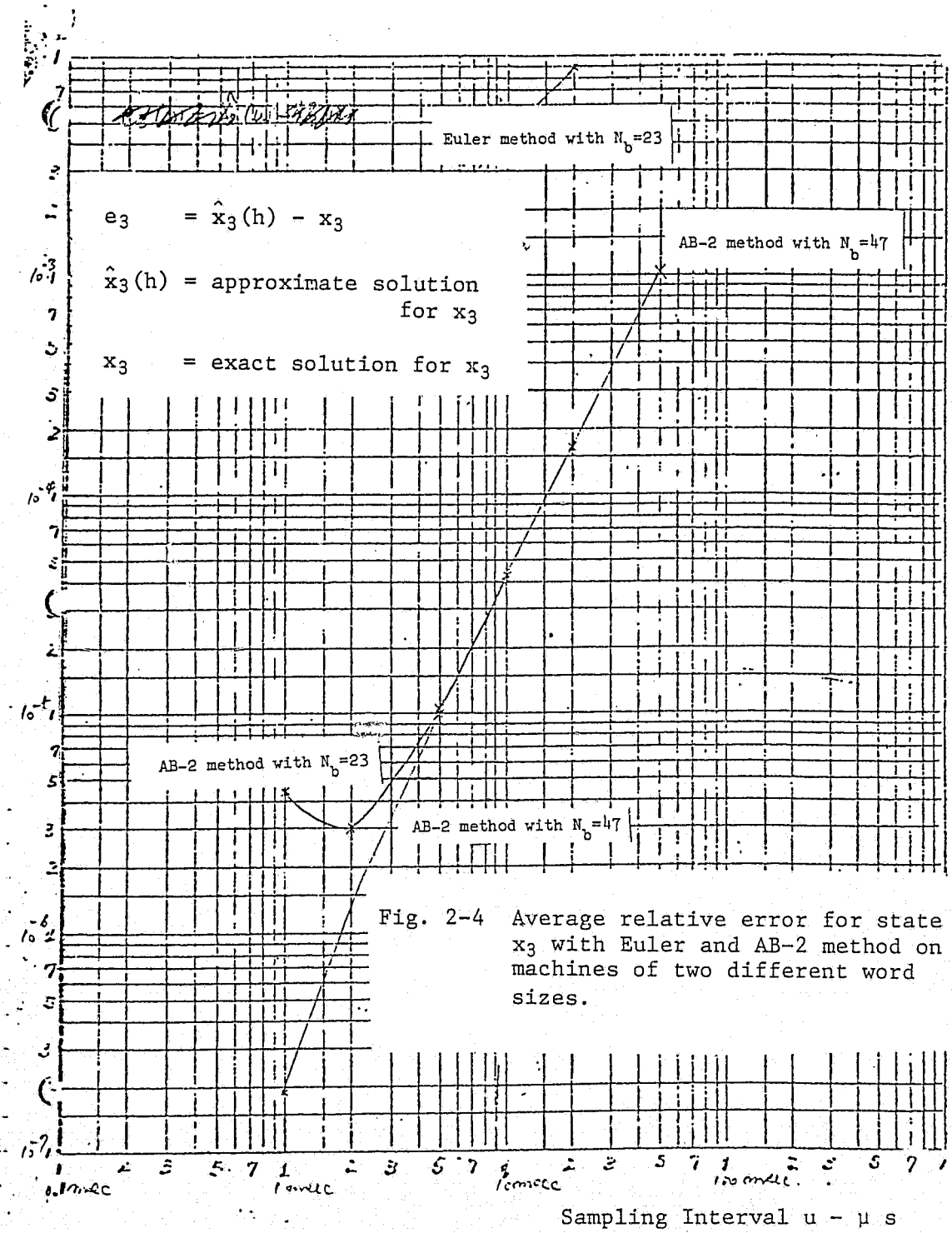


Fig. 2-4 Average relative error for state x_3 with Euler and AB-2 method on machines of two different word sizes.

somewhat smaller than, the value for $|h\omega_c|$ at the intersection. This seems to be explained, however, by the fact that ε_r is the maximum round-off error and the round-off error can be any value between zero and ε_r at random. Figure 2-1 also explains why the responses with the AB-2 method on machines with two largely different word sizes have almost same accuracy above h ≅ 5 msec.

Particular attention has been paid to the case using a large computer with the AB-2 method and h = 1/32 sec, because it was reported [8, 9] that this yields real-time flight simulation of satisfactory accuracy except for high-performance aircraft with high angular rates. Figures 2-2, 2-3, and 2-4 show that the average total propagated error with the AB-2 method on a machine with 32-bit floating-point arithmetic with a sampling interval h - .002 sec is almost two orders of magnitude smaller than that with the AB-2 method on a digital machine having 60-bit floating-point arithmetic and h - 1/32 sec. That means that, if the sampling frequency can be increased to 500 H_3 by means of efficient parallel operation of microprocessors, the accuracy of the simulation can be significantly improved over the accuracy currently obtained in real-time flight simulation on a large digital computer.

The above discussion was for a single open-loop system. The next step in the research will include closed-loop operation and generalization to other systems. It has not yet been established that the AB-2 is a suitable method for parallel processing; so it must be studied, too.

REFERENCES

1. Parrish, E. A. Jr., Cook, G., and McVey, E. S., "Real-Time Flight Simulation Methodology," Semi-annual Report No. UVA/528085/EE76/103, August 1976.

2. Fletcher, R., and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization," Computer Jnl., Vol. 6, No. 2 (1963), pp. 163-68.

3. Athanassopoulos, J. A., and Warren, A. D., "Design of Discrete Time Systems by Mathematical Programming," Proc. 1968 Hawaii Int. Conf. Syst. Sci., University of Hawaii Press, Honolulu, 1968, pp. 224-227.

4. Burrus, C. S., and Parks, T. W., "Time Domain Design of Recursive Digital Filters," IEEE Trans. Audio Electroacoust., Vol. AU-18, No. 2, pp. 137-141, June 1970.

5. Steiglitz, K., "Computer Aided Design of Recursive Digital Filters," IEEE Trans. Audio Electroacoustic, Vol. AU-18, No. 2. (June 1970), pp. 123-129.

6. Kuo, B. C., "Analysis and Synthesis of Sampled Data Control Systems," Prentice-Hall, 1963.

7. Beck, M. S., "Adaptive Control--Fundamental Aspects and Their Application," Proc. of 1st Anl. Advanced Control Conf. (Dun-Donnelly Publishing, Purdue University, April 29-May 1, 1974).

8. Wilson, J. W., and Steinmetz, G. G., Analysis of Numerical Integration Techniques for Real-Time Digital Flight Simulation, NASA TN D-4900, 1968.

9. Barker, L. E., Bowles, R. L., and Williams, L. H., Development and Application of a Local Linearization Algorithm for the Integration of Quaternion Rate Equations in Real-Time Flight Simulation Problems, NASA TN D-7347, 1973.